5-3-2016

A Modification and Application of Parametric Continuation Method to Variety of Nonlinear Boundary Value Problems in Applied Mechanics

Akshay Patil
axp6614@rit.edu

A Modification and Application of Parametric Continuation Method to Variety of Nonlinear Boundary Value Problems in Applied Mechanics

by

Akshay Patil

A thesis submitted to the faculty of Rochester Institute of Technology at Rochester in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

May 3rd 2016

Advisor: Dr. Alexander Liberson

Committee Member: Dr. Amitabha Ghosh

Committee Member: Dr. Panchapakesan Venkataraman

Department Representative: Dr. Aganemnon Crassidis

KATE GLEASON COLLEGE OF ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

Committee Approval:

_____

Dr. Alexander Liberson, Associate Professor

Thesis Advisor, Department of Mechanical Engineering

_____

Dr. Amitabha Ghosh, Professor

Committee Member, Department of Mechanical Engineering

_____

Dr. Panchapakesan Venkataraman, Associate Professor

Committee Member, Department of Mechanical Engineering

_____

Dr. Aganemnon Crassidis, Professor

Department Representative, Department of Mechanical Engineering

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Dr. Alexander Liberson for the continuous support of my Master's study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Amitabha Ghosh, Dr. Panchapakesan Venkataraman, and Dr. Aganemnon Crassidis, for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

Throughout my time at RIT, I have had the opportunity to work with and learn from some of the greatest minds that I have had the pleasure to encounter; the faculty of the Mechanical Engineering Department especially Dr. Stephen Boedo, Dr. Satish Kandlikar, Dr. Kathleen Lamkin-Kennard, and Dr. William Humphrey. Their mentorship and guidance has had a significant impact of my experience as a student here.

Finally, I must express my very profound gratitude to my parents and to my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

ABSTRACT

In the field of engineering, researches often come across strong nonlinear boundary value problems which cannot be solved easily. Numerical convergence for many problems, typically solved by the Newton-Raphson linearization algorithm, is sensitive to the initial approach, relaxation parameters and differential topology. Emphasis in the present work is placed on the alternative approach, the so called parametric imbedding of a particular problem into the family of problems. While this may appear to complicate rather than to simplify the problem, its justification lies in the fact that a relation between infinitesimally close neighboring processes results in a simple Cauchy problem with respect to the introduced parameter.

Many problems in applied mechanics are reduced to the solutions of systems of nonlinear algebraic, transcendental, differential or integral-differential equations containing an explicit parameter. These are problems in the areas of thermo-fluids, gas dynamics, deformable solids, heat transfer, biomechanics, analytical dynamics, catastrophe theory, optimal control and others. A parameter found in these models is not unique, and may be easily identified as a load which could be geometric, structural, and physical or it could be introduced artificially. An important aspect of these problems is a question of the variation of the solution when parameter is incrementally changed.

The growing interest in nonlinear problems in engineering has been intensified by the use of digital computers. This paved a way in development of the solution procedures which can be applied to a large class of nonlinear problems containing a parameter. An important aspect of these problems is the variation of the solution of with the parameter. Hence, method of continuing the solution with respect to the parameter is a natural and universal tool for the

analysis. It was originally introduced by Ambarzumian and Chandrasekar, and intensively studied by Bellman, Kalaba and others. Different problems of applied mechanics and physics with dominant nonlinearities due to convective phenomena, constituent models, finite deformation, bifurcation and others are analyzed and solved in the present work. The choice of the optimal continuation parameter, which ensures the best conditioning of the corresponding system of nonlinear equations, is discussed. Some modifications for stiff systems of ordinary nonlinear differential equations are suggested and applied. Effectiveness of the continuation method is demonstrated by comparing the results with the stiff boundary value problem numerical solvers implemented using commercial softwares. The objective of the research is to investigate applicability of the method as a universal approach to the wide range of nonlinear boundary value problems in different areas of mechanics: nonlinear mechanics of solids, bifurcation problems, Newtonian and Non-Newtonian fluids, thermo-fluids, gas-dynamics, control, inverse problems.

.

Table of Contents

Nomenclature 6

List of Figures 7

List of Tables 8

1. PROBLEM INTRODUCTION 9

2. THE RESEARCH QUESTION 9

3. LITERATURE REVIEW 10

4. OBJECTIVES OF THE STUDY 12

5. BASIC THEORY DESCRIPTION AND STATEMENTS OF THE ANALYSING PROBLEMS 13

5.1 Nonlinear problems described by algebraic or transcendental equations 13

5.2 Nonlinear problems described by the system of algebraic or transcendental equations 16

5.3 Examples of applying different forms of CBP method 24

5.4 Nonlinear problems described by the system of ODE 36

5.5 An Alternative Formulation of Bellman's Method of Invariant Imbedding 54

6. CONCLUSION 57

7. REFERENCES 58

A. APPENDIX 62

Nomenclature

p - Continuation Parameter

x – Variable

τ – Alternative continuation parameter used to match Newton-Raphson method

F – Vector Function

X – Vector argument

J – Jacobian matrix

P – Continuation Parameter for system of algebraic and transcendental equations

σ - Continuation Parameter for Arc-Length method/length of the solution curve

E – Young's modulus

A – Cross-sectional Area

J – Polar moment of inertia

N – Compressive force

N_{cr}- Critical Compressive force

w_0/w – Initial and maximum displacement in the bar

K – Equilibrium constant

μ & a – Blend co-efficients

η – Similarity variable

υ - Kinematic viscosity

List of Figures

Figure 1: Vorovich and Zipalova's continuation parameter as the length σ of the solution curve K……..……21

Figure 2: Application of CBP method……..……25

Figure 3: Application of Newton-Raphson method…….....25

Figure 4: Von Misses Truss…….……...26

Figure 5: The numerical solution of the Von Mises' Truss showing the distribution of vertical displacement and the total potential energy as a function of load..…..…….29

Figure 6: The solution of the Von Mises' Truss showing the variation in vertical displacement with the applied force as a function of initial imperfection in each bar.……...32

Figure 7: Convergence of the equilibrium mole coefficients……35

Figure 8: Boundary layer over the plate……36

Figure 9: Velocity Profile of a Blasius Solution……41

Figure 10: Boundary Layer Flow around the Wedge……42

Figure 11: Velocity Profile across the distance to the wall for Faulkner-Skan solution…..….…43

Figure 12: Flow over a rotating disc…….……....44

Figure 13: The plot (top) shows the comparison for velocity distribution between the results obtained by the proposed method (left) and the results obtained from Schlichting……..46

Figure 14: The figure shows the solution of Troesch's equation using Wolfram Mathematica for n=1, n=2, n=3, and n=4 respectively. Upon further increasing the n, the solution achieves singularity……..50

Figure 15: The figure shows the solution of Troesch's equation using Matlab for n=1 and n=2 respectively. Upon further increasing the n, the solution achieves singularity…….52

Figure 16: The figure shows the solution of Troesch's equation using continuation by parameter method for varying n……..53

Figure 17: The figure shows the solution of Troesch's equation using invariant imbedding method for n=12…….56

List of Tables

Table 1: Shows the comparison of the values by obtained by E.M. Sparrow and J.J. Gregg and the values obtained by the proposed CBP method…….46

1. PROBLEM INTRODUCTION

The method of continuation by parameter (CBP method) has been found extremely useful in various fields of mechanics, physics and engineering. The objective of the present work is to analyze the CBP method and different forms derived from this method and make recommendation of their application to different branches of applied mechanics: nonlinear mechanics of solids, bifurcation problems, fluids, thermo-fluids, gas-dynamics, control, inverse problem of mechanics. Recommendations are given to optimize the choice of a parameter of continuation in different problems, including stiff problems.

2. THE RESEARCH QUESTION

To solve a nonlinear boundary value problem, iterative quasilinearization based on Newton's method, is typically used. In the vicinity of a solution Newton's method converges quadratically and the method can produce results very efficiently. However, the rate of convergence is typically small (unless the function is quadratic), so that the benefits of a high rate of convergence is difficult to obtain. Typically convergence is very sensitive to initial approach of the method, relaxation parameters and topology of the relating function. We consider the potential application of parametric continuation as a universal method of solving nonlinear boundary value problems in different areas of mechanics, physics and applied mathematics. Some of the problems introduced in this study are new and have not been solved before by the method of continuation. Special questions such as the optimal choice of continuation parameter, and application of the method to stiff boundary value problems, are discussed.

3. LITERATURE REVIEW

The concept of CBP was applied to the transformation of a boundary value problem to a Cauchy problem by Ambarzsumian [1] and Chandrasekhar [2], and generalized by Bellman and Kalaba [3], [4]. This method was found extremely useful in various fields of physics, like a neutron transport theory [5], [9], radiative transfer [2], [5], random walk and scattering [6], wave propagation [6], [7], rarefied gas dynamics [3], Hamilton's equation of motion [8], and the flow in chemical reactors [10]. A fairly complete bibliography prior to 1962 can be found in the books by Wing [9] and Bellman *et al.* [5]. For more recent works, the books by Lee [10], Meyer [11], Scott [12] and Na [13] can be consulted. In the years since then a large number of new studies has been developed and applied to the problems. Recently published book of Grigoluyk *et al.* [14] presents application of a CBP method and a contemporary literature to the nonlinear problems in solid mechanics.

The surveys [5], [9], [10], [15] have been compiled decades ago. Our purpose is to review and systemize the latest applications of the CBP method by drawing attention to the related benefits and difficulties of the applications, along with the means to overcome these difficulties.

The basic CBP method is characterized by the presence of a parameter in the boundary value problem (BVP) which may be expressed functionally as,

$$f(x,p) = 0, \text{ where p is the parameter}$$

Assuming that initial value of a parameter (usually zero) simplifies the original BVP such that the corresponding solution becomes trivial, and the final value of the parameter (typically - one) results in a BVP required to solve, the overall problem can be reduced to a Cauchy's initial value problem stepwise integration procedure along the introduced parametric curve.

The choice of an appropriate continuation parameter is one of the crucial aspects of successful continuation. Vorovich and Zipalova [19], and almost at the same time Ricks [20], [21] raise the question of choosing the direction of continuation which ensures the best conditioning of the solution of the linearized system of NxN equations (Jacobian). As a result it is shown that the best direction corresponds to the parametric curve in N+1 dimensional space, created by N dimensional solution vector and originally chosen parameter p. The geometrical interpretation with applications to different problems of nonlinear mechanics is given in [22], [23], [25].

One of the first comparative studies of different continuation methods is presented in [26], [27], where the simplest explicit Euler type continuation method was compared with Newton-Raphson, Runge-Kutta and others [24]. The Mises three-hinged arch was used as a test problem. It was found that the number of steps in the stepwise CBP method was much less dependent on the magnitude of the final displacement than prescribed other methods. The majority of authors give a preference to the explicit scheme in a predictor step along with a correction of the solution using some appropriate corrector step [17], [18]. The question of a step length is as a rule concluded by numerical experiments.

Application of CBP to the Navier-Stokes equation is reported in [28]. The essential idea is to compute solutions when parameter – Reynolds number, is varying along a continuation path. It was noted that the Jacobian matrix reduces its ill conditioning during iterations. To overcome the problem, the arch-length continuation technique, described in [16], [29] is used. Numerical results were obtained for the moving lid driven cavity and the Bratu reaction-diffusion problem.

4. OBJECTIVES OF THE STUDY

The objective of the research is to:

- Investigate applicability of the method as a universal approach to the wide spectrum of nonlinear boundary value problems in different areas of mechanics and engineering
- Compare different forms of the CBP methods
- Optimize the choice of a continuation parameter for the nonlinear regular and stiff problems

5. BASIC THEORY DESCRIPTION AND STATEMENTS OF THE ANALYSING PROBLEMS

5.1 Nonlinear problems described by algebraic or transcendental equations

5.1.1 Basic approach

To introduce the idea consider the algebraic equation, where x is unknown, and p is a parameter

$$f(x,p)=0$$

Assume $x= x(p)$ is a monotonic and a smooth function. By differentiating with respect to parameter p, arrive to

$$\frac{\partial f}{\partial p}+\frac{\partial f}{\partial x}\frac{dx}{dp}=0$$

wherefrom,

$$\frac{dx}{dp}=-\frac{\frac{\partial f}{\partial p}}{\frac{\partial f}{\partial x}}$$

The obtained equation enables to formulate the Cauchy problem for determining $x(p)$ with initial condition $x_0=x(p_0)$. To construct solution this approach opens up the possibility of using various well studied integration schemes for initial value algorithm. The simplest, - the Euler's method, leads to the following iterative algorithm

$$x_{i+1} = x_i - \left[\frac{\partial f}{\partial p} \Big/ \frac{\partial f}{\partial x} \right]_i \Delta p + O(\Delta p^2), \qquad \Delta p = p_{i+1} - p_i \tag{1}$$

It is not difficult to construct algorithms for other schemes having a higher order of accuracy, such as Runge-Kutta, Adams or modified Euler methods. We will be focusing on principle features of CBP methods, increasing if needed, the discrete cell numbers to match the known canonical solutions, obtained using different methods.

It is interesting to compare CBP method algorithm with the classical Newton-Raphson algorithm, which being applied to the original problem looks as the following

$$x_{i+1} = x_i - \left[f(x) \Big/ \frac{\partial f}{\partial x} \right]_i + O(\Delta x^2) \tag{2}$$

Easy to see that the CBP and Newton-Raphson algorithms are not identical, and could be quite different depending on the choice of the parameter of continuation '*p'*.

5.1.2 Specific form of a parameter of continuation equivalent to the Newton-Raphson procedure

Before we conclude, we mention some specific possibility of using the CBP method for the nonlinear equation,

$$h(x) = 0 \tag{1}$$

Let $x_0$be the starting approximation. We construct an equation with a parameter as follows

$$f(x,p) = h(x) - (1-p)h(x_0) \qquad (2)$$

Here the parameter p is introduced so that x_0 is a solution when p=0. And when p=1 the equation transforms into the original one. A new the parameter τ is introduced so as to obtain the solution in Newtonian form,

$$1 - p = e^{-\tau}, \quad \tau \in [0, \infty]$$

The equation becomes

$$f(x,\tau) = h(x) - e^{-\tau}h(x_0) \qquad (3)$$

Differentiation with respect to parameter τ results in

$$\frac{\partial h}{\partial x}\frac{\partial x}{\partial \tau} + e^{-\tau}h(x_0) = 0$$

Wherefrom

$$\frac{\partial x}{\partial \tau} = -\frac{e^{-\tau}h(x_0)}{\frac{\partial h}{\partial x}} = -\frac{h(x)}{\frac{\partial h}{\partial x}} \qquad (4)$$

which leads to the Newton –Raphson method with the step $\Delta\tau = 1$,

$$x_{i+1} = x_i - \left[h(x) \Big/ \frac{\partial h}{\partial x} \right]_i \qquad (5)$$

This specific choice of a continuation parameter results in the classical Newton-Raphson method.

5.2 Nonlinear problems described by the system of algebraic or transcendental equations

5.2.1 Basic approach

Consider a system of m nonlinear equations with m unknowns, containing a parameter P

$$F_i(X_1, X_2, \ldots, X_m, P) = 0, \quad i = 1, 2, \ldots, m \tag{1}$$

Using vector notations,

$$X = [X_1, X_2, \ldots, X_m]^T \tag{2}$$

$$F = [F_1, F_2, \ldots, F_m]^T \tag{3}$$

we present the system of equations in a vector form

$$F(X, P) = 0 \tag{4}$$

Introducing Jacobian matrix

$$J = \frac{\partial F}{\partial X} = \frac{\partial (F_1, F_2, \ldots, F_m)}{\partial (X_1, X_2, \ldots, X_m)} = \left[\frac{\partial F_i}{\partial X_j}\right] \tag{5}$$

and differentiating the vector system of nonlinear equation, we arrive at

$$J \frac{\partial X}{\partial P} + \frac{\partial F}{\partial P} = 0, \tag{6}$$

The following steps proceed until continuation parameter ‘p’ reaches the final value, which is p=1.

$$X_{(i+1)} = X_{(i)} - J^{-1}\left(X_{(i)}, P_i\right)F_P\left(X_{(i)}, P_i\right)\Delta P, \qquad i = 1, 2, \ldots, n \qquad (8)$$

where,

$$\Delta P = P_{i+1} - P_i, F_P = \frac{\partial F}{\partial P} \qquad (9)$$

Analogously to the scalar case, analyzed in 5.1.1, the obtained equation looks different from the Newton-Raphson procedure applied to the nonlinear system of equations.

5.2.2 Application of CBP to the system of nonlinear differential equations

Consider application of the CBP method to the nonlinear system of differential equations. For the purpose of illustration consider the system of 2 non-linear ordinary differential equations:

$$\left.\begin{aligned} u' &= f(u, v) \\ v' &= g(u, v) \\ u(0) &= u_0; \quad v(1) = v_1 \end{aligned}\right\} \qquad (1)$$

Parameter ‘p’ can be introduced in multiple ways, for instance,

$$\left.\begin{aligned} u' &= f(u, v) \\ v' &= g(u, v) \\ u(0) &= pu_0; \qquad v(0) = pv_1 \end{aligned}\right\} \qquad (2)$$

It’s easy to see that at p=0, u=v=0 is a trivial solution.

Assume ‘u’ and ‘v’ are continuous functions of p. We are going to create the Cauchy problem across parameter ‘p’, when $u(p=0)$ and $v(p=0)$ are the known values and $u(p=1)$ and $v(p=1)$ are the required solutions.

Introducing derivatives (sensitivities by p),

$$\bar{u} = \frac{\partial u}{\partial p} \text{ and } \bar{v} = \frac{\partial v}{\partial p}$$

Differentiating equation (2) by p,

$$\begin{cases} \overline{u}' = \frac{\partial f}{\partial u}\bar{u} + \frac{\partial f}{\partial v}\bar{v}\,; \quad \overline{u}(0) = u_0 \\ \overline{v}' = \frac{\partial g}{\partial u}\bar{u} + \frac{\partial g}{\partial v}\bar{v}\,; \quad \overline{v}(0) = v_1 \end{cases} \tag{3}$$

Equation (3) with respect to sensitivities $\overline{u}$ and $\overline{v}$ is linear. We apply reduction to Cauchy method, presenting solution $\overline{u}, \overline{v}$ as a result of superposition.

$$\begin{cases} \overline{u} = \overline{u^1} + \mu\overline{u^2} \\ \overline{v} = \overline{v^1} + \mu\overline{v^2} \end{cases} \tag{4}$$

Each component $\overline{u^1}$, $\overline{v^1}$ and $\overline{u^2}$, $\overline{v^2}$ is the solution of a Cauchy problem

$$\overline{u^1}(0) = u_0; \quad \overline{v^1}(0) = 0$$

$$\overline{u^2}(0) = 0; \quad \overline{v^2}(0) = 1$$

Introduce vectors $z_1 = \begin{pmatrix} \overline{u^1} \\ v^1 \end{pmatrix}$ and $z_2 = \begin{pmatrix} \overline{u^2} \\ v^2 \end{pmatrix}$, so that

$$\frac{dz_1}{dx} = Az_1 \qquad z_1(x=0) = \begin{pmatrix} u_0 \\ 0 \end{pmatrix}$$

$$\frac{dz_2}{dx} = Az_2 \qquad z_2(x=0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Each equation is integrated using implicit numerical scheme,

$$\frac{z^{k+1} - z^k}{h} = Az^{k+1} \rightarrow z^{k+1} = [I - hA]^{-1} z^k$$

(h-a finite difference step, z^k-unknown vector associated with the Kth node)

As a result, total solution is composed based on superposition

$$z = z^1 + \mu z^2$$

$$\text{Or} \quad \begin{pmatrix} \overline{u} \\ \overline{v} \end{pmatrix} = \begin{pmatrix} \bar{u}^1 \\ \bar{v}^1 \end{pmatrix} + \mu \begin{pmatrix} \bar{u}^2 \\ \bar{v}^2 \end{pmatrix} \qquad (5)$$

This satisfies to original ODE and left boundary condition.

To satisfy to the right boundary condition choose μ according to,

$$\bar{v}(1) = \bar{v}^1(1) + \mu \bar{v}^2(1) = v_1 \rightarrow \mu = \frac{v_1 - \bar{v}^1(1)}{\bar{v}^2(1)} \qquad (6)$$

The current elementary p-steps is processed as

$$u(p+\Delta p) = u(p) + \bar{u}(p).\Delta p$$
$$v(p+\Delta p) = v(p) + \bar{v}(p).\Delta p \quad \Big\} \ (7)$$

Procedure is continued until p =1.

5.2.3 Arch-Length Continuation (Vorovich and Zipalova [19], Ricks [21])

According to Vorovich, Zipalova [19] and Rick [21], parameter 'p'is introduced such that

$$\overrightarrow{X}(p = p_0) = \overrightarrow{X_0} \qquad (1)$$

Another parameter 'σ' is introduced as a new continuation parameter. Assuming $X_i = X_i(\sigma)$ and $p = p(\sigma)$ equation (1) from section 5.2.1 is differentiating by the new parameter 'σ'.

$$\sum_{i=1}^{m} \frac{\partial F_i}{\partial X_i}\frac{dX_j}{d\sigma} + \frac{\partial F_i}{\partial P}\frac{dP}{d\sigma} = 0,\ i = 1, \dots, m; \qquad (2)$$

The system of 'N' equations contains (N+1) unknown variables,

$$\frac{dX_i}{d\sigma}, i = 1,2, \dots, m; and\ \ \frac{dP}{d\sigma}. \qquad (3)$$

The supplemental equation

$$\sum_{i=1}^{m} \left(\frac{dX_i}{d\sigma}\right)^2 + \left(\frac{dP}{d\sigma}\right)^2 = 1 \qquad (4)$$

being added to the system (2), allows us to calculate all the sensitivities $\frac{dX_i}{d\sigma}, i = 1,2, \dots, m; and\ \ \frac{dP}{d\sigma}$

Figure 1 illustrates arc length continuation strategy in a three-dimensional space: ***(X₁, X₂, P)***

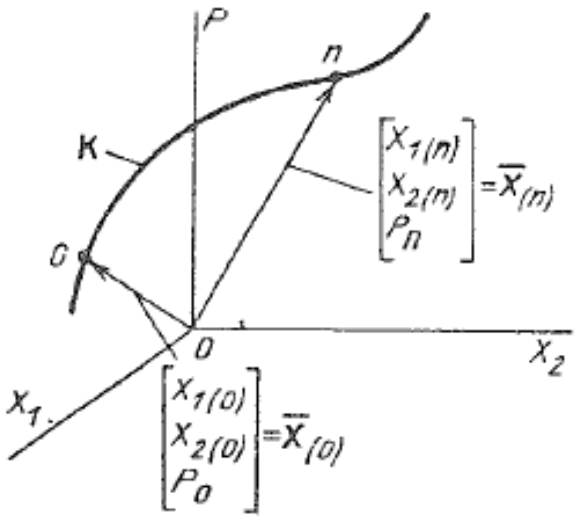


Figure 1: Vorovich and Zipalova's arc length continuation strategy

Ricks [22] proved that the choice of a parameter 'σ', as the length of a solution curve, ensures the best conditioning of the solution of the corresponding system of linear equations.

Introducing the following matrix and vector – column notations:

$$A = \left[\frac{\partial F_i}{\partial X_j}\right], \quad F_P = \left[\frac{\partial F_i}{\partial \sigma}\right], \quad \bar{X} = \left[\frac{\partial X_i}{\partial \sigma}\right]; \quad \bar{P} = \frac{\partial P}{\partial \sigma}$$

the basic system can be transformed to the compact form

$$A\bar{X} = -F_P\bar{P} \qquad (3)$$

$$\bar{X}^T\bar{X} + \bar{P}^2 = 1 \qquad (4)$$

To solve the system of equations with respect to $\bar{X}$, introduce vector $\bar{X}_1$, satisfying to

$$A\bar{X}_1 = -F_P \qquad (5)$$

Then $\bar{X} = \bar{X}_1\bar{P}$, and $\bar{P}$ follows from quadratic equation

$$\bar{P} = \frac{1}{\sqrt{1+\bar{X}_1^T\bar{X}_1}} \qquad (6)$$

The corresponding algorithm looks as the following:

1. Set *i=0* and let. X_0 be the known solution at *P=0.*
2. Based on X_0 calculate $A = \left[\frac{\partial F_i}{\partial X_j}\right], \quad F_P = \left[\frac{\partial F_i}{\partial \sigma}\right]$
3. Find vector $\bar{X}_1$ solving equation $A\bar{X}_1 = -F_P$
4. Find optimized parameter $\bar{P} = \frac{1}{\sqrt{1+\bar{X}_1^T\bar{X}_1}}$ and vector $\bar{X} = \bar{X}_1\bar{P}$
5. Set *i=i+1.* Calculate X_{i+1}=$X_i + \bar{X}_1\Delta\sigma$ and P_{i+1}=$P_i + \bar{P}_1\Delta\sigma$
6. Go to step 1, until reaching the nominal value for P

5.2.4 Simplified Arch-Length Continuation (P=0)

Here we analyze the case when parameter P could be omitted from consideration, Let us consider for simplicity the two-dimensional problem

$$F(X_1, X_2) = 0 \qquad (1)$$

Our objective is to find $X_2 = f(X_1)$ for a variety of X_1.

According to Vorovich-Zipalova (see 5.2.2), the optimal continuation parameter is the length σ of the solution curve K**,** Figure 1, on a plane (X_1, X_2). In the absence of a parameter P,

$$d\sigma^2 = dX_1^2 + dX_2^2, \qquad \text{or} \qquad \bar{X}_1^2 + \bar{X}_2^2 = 1, \qquad \bar{X}_1 = \frac{dX_1}{d\sigma}; \quad \bar{X}_2 = \frac{dX_2}{d\sigma} \tag{2}$$

On differentiation the function *F* with respect to σ, obtain $\left(F_1 = \frac{\partial F}{\partial X_1}, \ F_2 = \frac{\partial F}{\partial X_2}\right)$

$$F_1\bar{X}_1 + F_2\bar{X}_2 = 0 \tag{3}$$

$$\bar{X}_1^2 + \bar{X}_2^2 = 1 \tag{4}$$

Solution of these equations can be presented in the form

$$\bar{X}_1 = \frac{F_2}{\sqrt{F_1^2+F_2^2}}; \qquad \bar{X}_2 = \frac{-F_1}{\sqrt{F_1^2+F_2^2}} \tag{5}$$

The corresponding algorithm looks as the following:

1. Set *i=0* and let. $X_1 \, and \, X_2 \, be \, the$ known solution at σ =*0.*
2. Specify increment of parameter Δσ
3. Based on $X_1 \, and \, X_2$ calculate $F_1 = \frac{\partial F}{\partial X_1}, \ F_2 = \frac{\partial F}{\partial X_2}$
4. Find sensitivities of solution to σ: $\bar{X}_1 = \frac{F_2}{\sqrt{F_1^2+F_2^2}}; \qquad \bar{X}_2 = \frac{-F_1}{\sqrt{F_1^2+F_2^2}}$
5. Set *i=i+1*. Calculate X_1=$X_1 + \bar{X}_1\Delta\sigma$; X_2=$X_2 + \bar{X}_2\Delta\sigma$.
6. Go to step 1, until reaching the upper limit value for σ.

5.3 Examples of applying different forms of CBP method

5.3.1 Bernoulli's lemniscate.

We consider the construction of Bernoulli's lemniscate by the CBP method. This is a complex curve in a form of a lying figure eight, comprising two loops, which makes it a good example to demonstrate effectiveness of CBP and its advantage over the classical Newton-Raphson method.

The equation of lemniscate in the x_1, x_2 axes is of the form

$$f(\mathbf{x}) = (x_1^2 + x_2^2)^2 - 2a^2(x_1^2 - x_2^2) = 0, \quad X = (x_1, x_2)^T \tag{1}$$

Using the simplified version of Vorovich-Zipalova's algorithm, we obtain

$$f_1\bar{x}_1 + f_2\bar{x}_2 = 0 \tag{2}$$

$$\bar{x}_1^2 + \bar{x}_2^2 = 1 \tag{3}$$

Where

$$f_1 = \frac{\partial f}{\partial x_1} = 4x_1(x_1^2 + x_2^2 - a^2) \tag{4}$$

$$f_2 = \frac{\partial f}{\partial x_2} = 4x_2(x_1^2 + x_2^2 + a^2) \tag{5}$$

$$\bar{x}_1 = \frac{\partial x_1}{\partial \sigma}; \quad \bar{x}_2 = \frac{\partial x_2}{\partial \sigma}$$

Solution of these equations is presented in the form

$$\bar{x}_1 \quad = \frac{f_2}{f_1^2+f_2^2}; \qquad \bar{x}_2 \quad = \frac{-f_1}{f_1^2+f_2^2} \tag{6}$$

Starting from the point $\sigma = 0, x_1 = a\sqrt{2}\,, x_2 = 0)$, we proceed iterations

$$x_1 = x_1 + \bar{x}_1 \Delta\sigma; \quad x_2 = x_2 + \bar{x}_2 \Delta\sigma \tag{7}$$

Figure 1 presents the results of integrating the Cauchy problem with the use of an explicit parameter 'σ'. Figure 2 presents the result obtained by integration according to the classical Newton-Raphson scheme. Unlike the CBP, the Newton-Raphson method is not capable to pass the points with a vertical tangent line where Jacobian turns to zero. The corresponding Matlab code is presented in the appendix.

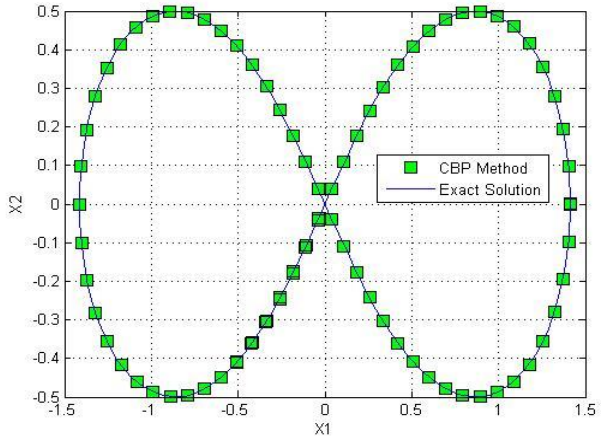


Figure 2: Application of CBP method (a=1)

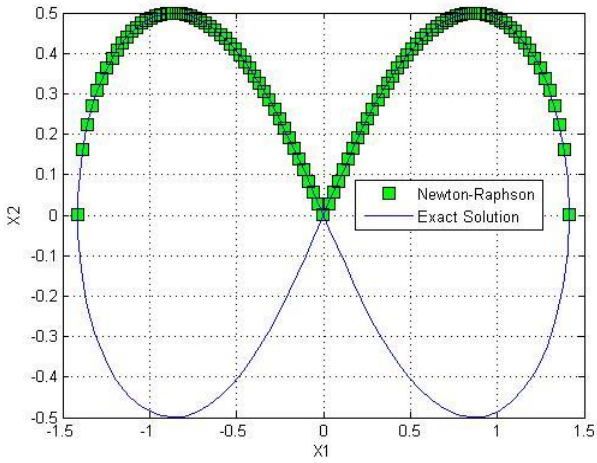


Figure 3: Application of Newton-Raphson method (a=1)

5.3.2 Stability of an imperfect Von Mises Truss

The Von Mises' truss is a classic example of instability of a geometrically nonlinear mechanical system comprised of perfectly straight elastic bars. The present investigation considers imperfect structures, where each bar is characterized by initial deviation from the straight line. Geometric notation is presented by the Fig.4. Let E – be the Young modulus of the material, w_0,w – initial and current maximum values of the bars' deviation from the straight direction; A,J – the area and moment of inertia of the cross section, N_{cr} – critical compressing (buckling) force, $N_{cr} = \pi^2 EJ/l^2$, N – the compressive force

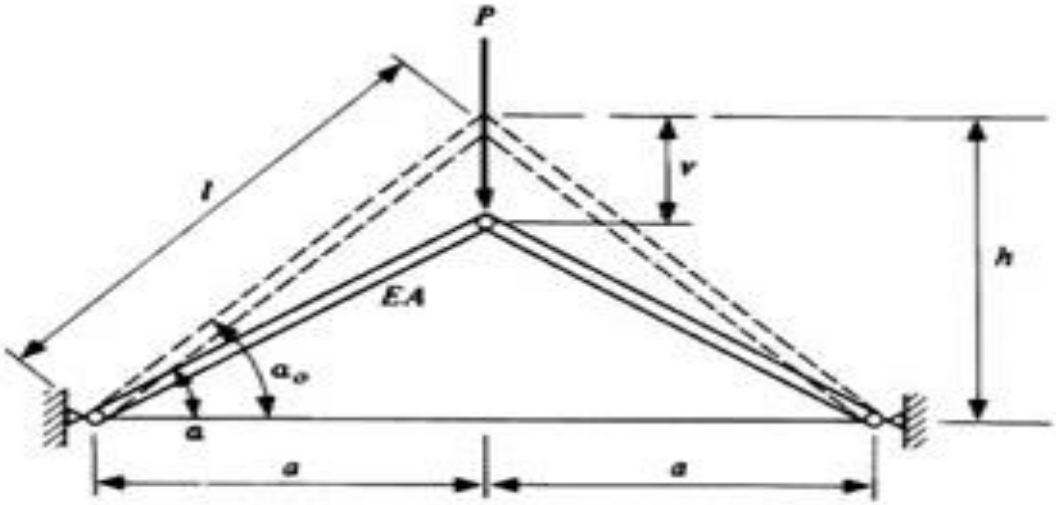


Figure 4: Von Misses Truss

The compatibility equation is of the form,

$$\frac{Nl}{EA} + \frac{\pi^2}{4l}(w^2 - w_0^2) = \frac{a}{cos\alpha_0} - \frac{a}{cos\alpha} \qquad (1)$$

The first term on the left of the equal sign quantifies the shortening of the bar due to the compressive force; the second term – shortening due to the bending of the initially geometrically imperfect bar. The term on the right is the overall change of the length, expressed via geometric parameters.

Introducing critical buckling force into the expression for the normal displacement w,

$w = \frac{w_0}{1-N/N_{cr}}$, arrive at the following transcendental equation

$$F(P, \propto) = \frac{Nl}{EA} + \frac{\pi^2 w_0^2}{4l}\left[\frac{1}{(1-N/N_{cr})^2} - 1\right] + \frac{a}{cos\alpha} - \frac{a}{cos\alpha_0} = 0 \qquad (2)$$

where,

$$N = P/(2sin \propto)$$

As it follows from the Figure 4, the vertical displacement of the joint of both bars can be presented as

$$v = a(tan \propto_0 - \tan \propto)$$

Introducing non-dimensional quantities

$$\bar{N} = \frac{N}{EA}; \quad \bar{P} = \frac{P}{EA}; \quad \bar{v} = \frac{v}{a}; \quad a_1 = \frac{\pi^2}{4}\left(\frac{w_0}{i}\right)^2; \qquad a_2 = \frac{Al^2}{\pi^2 J}; \qquad (3)$$

we can present basic equations in the following compact form

$$F(\bar{P}, \propto) = \bar{N} + a_1\left[\frac{1}{(1-a_2\bar{N})^2} - 1\right] + \frac{cos\alpha_0}{cos\alpha} - 1 = 0 \qquad (4)$$

$$\bar{N} = \bar{P}/(2sin \propto) \qquad (5)$$

$$\bar{v} = tan \propto_0 - \tan \propto \quad (6)$$

Using the simplified version of Vorovich-Zipalova algorithm, obtain

$$F_p \bar{P}_\sigma + F_\alpha \bar{\alpha}_\sigma = 0 \quad (7)$$

$$\bar{P}_\sigma{}^2 + \bar{\alpha}_\sigma{}^2 = 1 \quad (8)$$

where

$$F_p = \frac{\partial F}{\partial \bar{P}} = \frac{\partial F}{\partial \bar{N}} \frac{\partial \bar{N}}{\partial \bar{P}}; \quad F_\alpha = \frac{\partial F}{\partial \propto} = \frac{\partial F}{\partial \bar{N}} \frac{\partial \bar{N}}{\partial \propto} \quad (9)$$

$$\bar{P} = \frac{\partial \bar{P}}{\partial \sigma}; \quad \bar{\alpha}_\sigma = \frac{\partial \bar{\propto}}{\partial \sigma} \quad (10)$$

Solution of these equations is presented in the form

$$\bar{P}_\sigma = \frac{F_\alpha}{\sqrt{F_p{}^2 + F_\alpha{}^2}}; \quad \bar{\alpha}_\sigma = -\frac{F_p}{\sqrt{F_p{}^2 + F_\alpha{}^2}} \quad (11)$$

Starting from the point $\sigma = 0, P = 0$, $\propto = \propto_0$, proceed iterations

$$P = P + \bar{P}_\sigma \Delta\sigma; \quad \propto = \propto + \bar{\alpha}_\sigma \Delta\sigma \quad (12)$$

Figure 5 presents the results of integrating the Cauchy problem with the use of an explicit parameter 'σ'. The plot on the top represents the force versus vertical displacement curve which describes equilibrium conditions for the Mises truss. The plot at the bottom shows the total potential energy as a function of displacement.

The curve for force versus displacement has two extremes namely, the maxima and the minima which divides the curve in three characteristic sections. Figure 6 shows the plot of equilibrium conditions in terms of the force vs displacement for different initial displacements (w_0)

characterizing geometric non-perfectness . The corresponding MATLAB code is presented in the appendix.

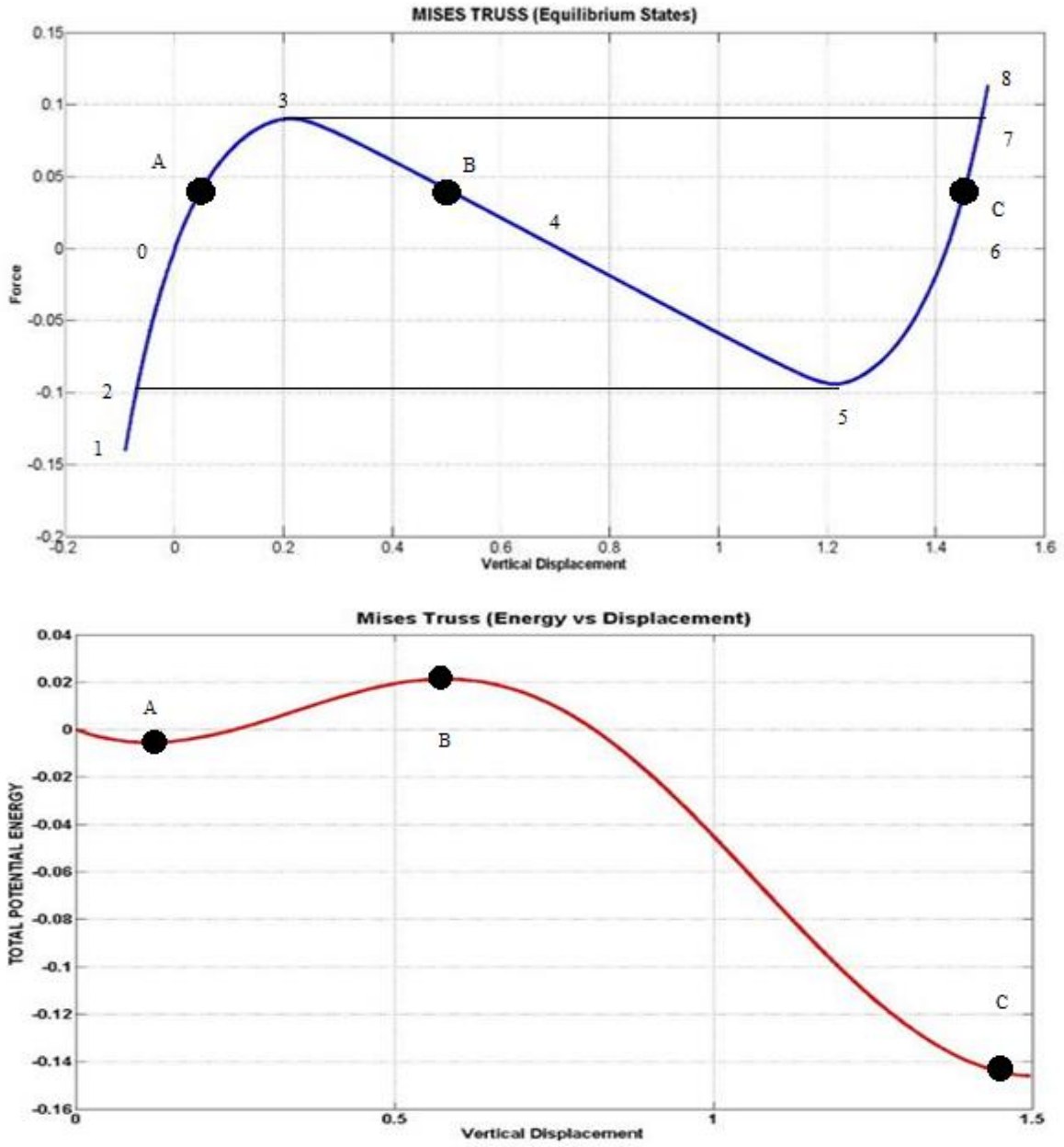


Figure 5: The numerical solution of the Von Mises' Truss showing variation of load P (top) and total potential energy (bottom) as functions of displacement

The loading process could be understood from the plot of Figure 5 describing equilibrium condition in terms of a load versus displacement. Let us now suppose that the loading process occurs monotonically from zero (the point zero). It is evident that after the system achieves the state characterized by the point 3, the system should execute a jump immediately to the section III (point 7) on the plot of Figure 5. Two different configurations of the system corresponds to one and the same load $P_3 = P_7$, i.e., immediately prior to the jump and immediately after it. Upon further increasing the force P, still higher points of the section III will correspond to equilibrium states of the system.

The process of unloading the system is described by the section III, which starts from some point 8 above point 7 on the same blue curve, and the system's equilibrium states will be characterized successively by the points 8-7-6-5. If the system is loaded to point 8 and then completely unloaded to the point 6, the system will not return to its original state. The negative (i.e. upward force) loads P will cause a transition of the system from 6 to 5. Further decreasing the force P (i.e. upon an increase in the load directed upward) the system will complete its reverse jump to the first section (point 2), and then a monotonic process of deformation is established downward along the section I.

The equilibrium points characterized by the points of section II and figure 5 are not realized in the entire process. We are convinced that unstable states correspond to these points. Now it is necessary to discuss the states of the system described by all the points of the horizontal straight line i.e. not only the equilibrium states but also the non-equilibrium states.

So the total potential energy of the deformed state consists of two parts [30]:

1) the potential energy of the deformation, which we can specify by equation

$$PE_1 = 2\frac{N^2 l}{2EA} = EAl\left[1 - cos\alpha_0\sqrt{tan^2\alpha_0 + \left(1 - \frac{\upsilon}{a}tan\alpha_0\right)^2}\right]^2 \quad (13)$$

2) the loading potential

$$PE_2 = -P\upsilon \quad (14)$$

Consequently, the system's total potential energy will be,

$$PE = PE_1 + PE_2 = EAl\left[1 - cos\alpha_0\sqrt{tan^2\alpha_0 + \left(1 - \frac{\upsilon}{a}tan\alpha_0\right)^2}\right]^2 - P\upsilon \quad (15)$$

The relationship between the total potential energy and the vertical displacement is shown in figure 5 with the red plot (bottom figure).

Now the stability of each point on the curve could be understood by the total potential energy curve. Suppose we are given 3 points A, B and C on the force versus displacement curve. These 3 points A, B and C corresponds to the same load 'P'. Now we plot the total potential energy curve for the corresponding load 'P' and position our points A, B and C accordingly. As per the theory suggests, the points with lower potential energy are stable whereas the points with higher potential energy are unstable. Now looking at the plot, we see points A and C have lower potential energy whereas B has a very higher potential energy. So we can say that points A and C are stable whereas point B is unstable.

The classical Newton-Raphson method being applied to the bifurcation Mises truss problems diverges, since Jacobian turns zero at the extreme points.

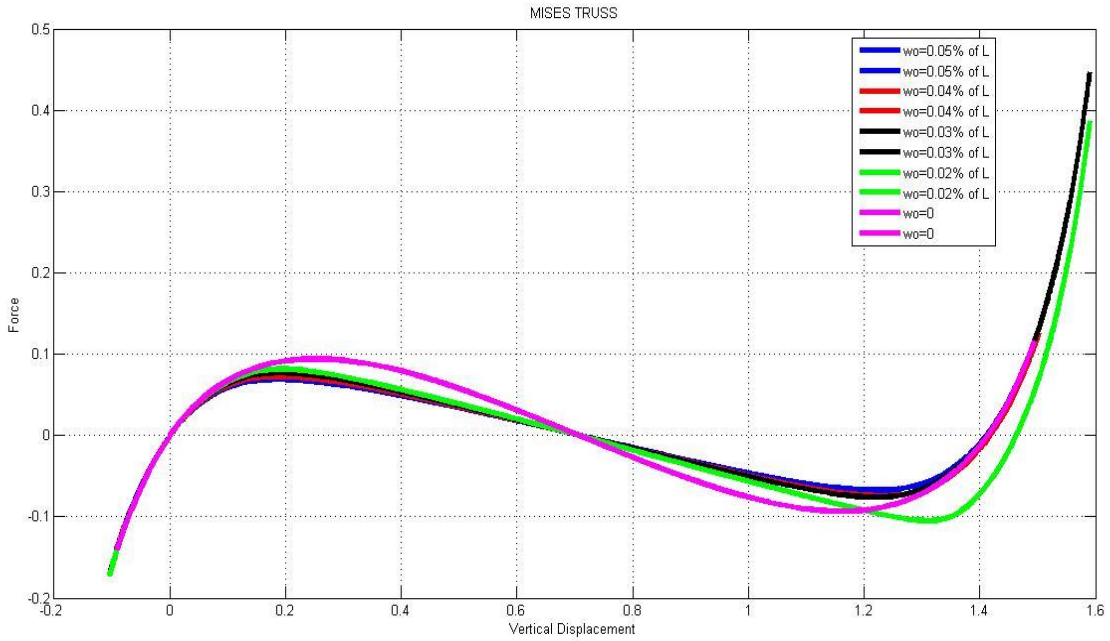


Figure 6: The solution of the Von Mises' Truss showing the variation of equilibrium curves with initial geometric imperfections ($w_0 \neq 0$) in each bar.

5.3.3 Equilibrium composition of a gas mixture at high temperature

The theoretical determination of equilibrium compositions in the flow of chemically reacting fluids is complicated by the fact that it is necessary to obtain the simultaneous solutions of a system of nonlinear algebraic equations. Let us consider a one mole of CO2 at 10 atm which is heated in a steady-flow constant pressure process. It is required to find the equilibrium compositions at a total pressure of 10 atm and a range of temperature from 1000 to 6000 K. The equilibrium composition is assumed to consist of CO2, CO, and O2, and is characterized by the unknown constants *α, β and γ* - the number of moles, as shown by the following balance equation

$$CO_2 \rightarrow \alpha CO_2 + \beta CO + \Upsilon O_2 \qquad (1)$$

For the balance equations we can write:

-according to the balance of element C:

$$\alpha + \beta = 1 \qquad (2)$$

-according to the balance of the element O

$$2\alpha + \beta + 2\Upsilon = 2 \qquad (3)$$

According to the statement of a law of mass action

$$\frac{\beta\sqrt{\Upsilon}}{\alpha}\frac{1}{\sqrt{\alpha+\beta+\Upsilon}}\sqrt{P_r} = K$$

Or

$$\beta^2\gamma P_r - K^2\alpha^2(\alpha+\beta+\gamma) = 0 \qquad (4)$$

Where K – is the equilibrium constant, P_r – total pressure, given by the attached table

Equilibrium Constant of Reaction as a Function of Temperature

$T(K)$	$\ln K$	$T(K)$	$\ln K$
1000	−23.535	4000	1.593
2000	−6.641	5000	3.191
3000	−1.117	6000	4.239

Introduce parameter of continuation p as the following

$$\propto +\beta - 1 = (1-p)(\propto_0 + \beta_0 - 1) \qquad (5)$$

$$2 \propto +\beta + 2\gamma - 2 = (1-p)(2 \propto_0 + \beta_0 + 2\gamma_0 - 2) \qquad (6)$$

$$\beta^2\gamma P_r - K^2 \propto^2 (\propto +\beta+\gamma) = (1-p)(\beta_0^2\gamma_0 P_r - K^2 \propto_0^2 (\propto_0 + \beta_0 + \gamma_0) \qquad (7)$$

Where mole numbers marked by zero indices, are the relating initial approach, all chosen as 0.5

Differentiating by parameter, arrive to the following system

$$AV = R \qquad (8)$$

where,

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 1 & 2 \\ -K^2(3 \propto^2 + 2 \propto \beta + 2 \propto \gamma) & 2\gamma\beta P_r - K^2 \propto^2 & \beta^2 P_r - K^2 \propto^2 \end{bmatrix} \qquad (9)$$

$$R = -[\propto_0 + \beta_0 - 1 \quad (2 \propto_0 + \beta_0 + 2\gamma_0 - 2) \quad \beta_0^2\gamma_0 P_r - K^2 \propto_0^2 (\propto_0 + \beta_0 + \gamma_0)]^T \qquad (10)$$

$$V = \left[\frac{d\propto}{dp} \quad \frac{d\beta}{dp} \quad \frac{d\gamma}{dp}\right]^T \tag{11}$$

Starting from the point $p = 0, P_r = 0$, $\propto = \beta = \gamma = 0.5$, proceed iterations until $p = 1$

$$V = V + A^{-1}R\Delta p;$$

Figure 7 tracks evolution of mole coefficients during integration of the Cauchy problem with the use of an explicit parameter 'p'. The final results are practically the same obtained in [13]. The relating MATLAB based text code is presented in the attachment.

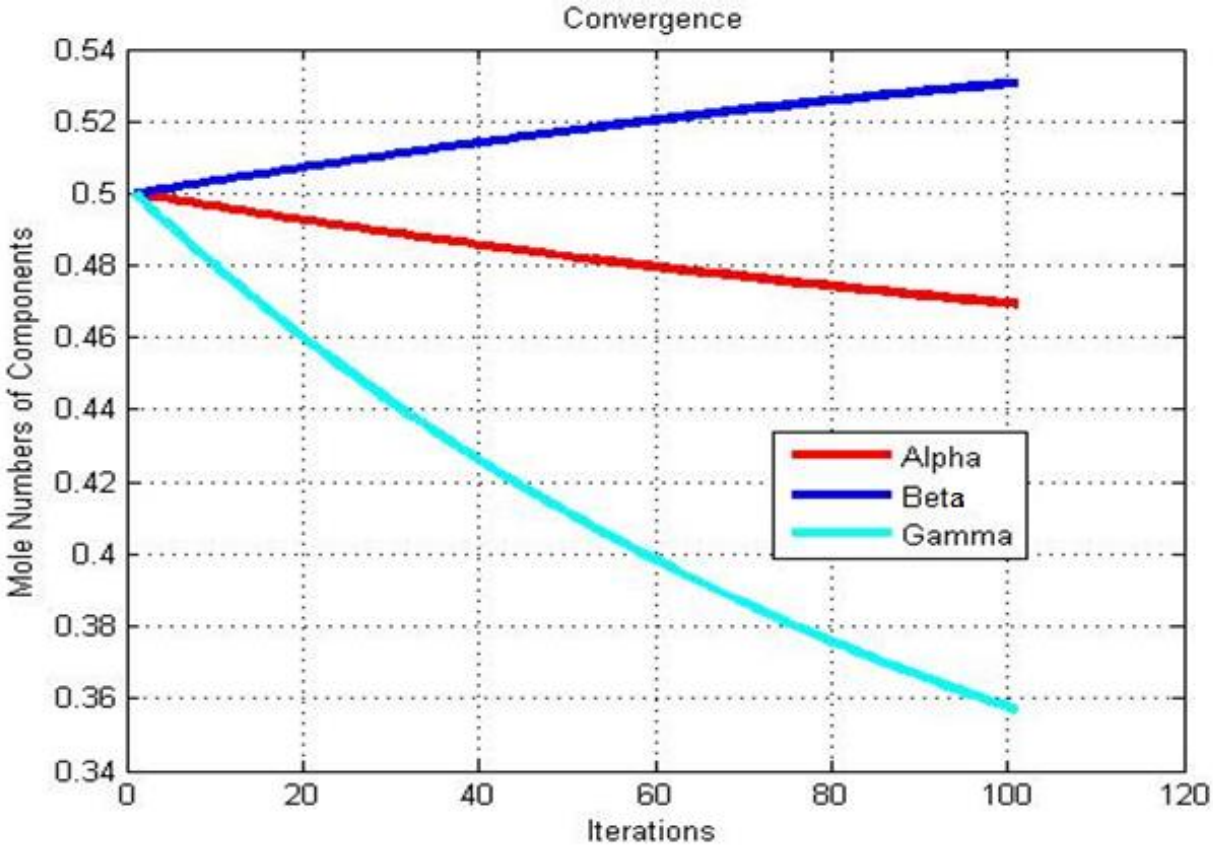


Figure 7: Convergence of the equilibrium mole coefficients

5.4 Nonlinear problems described by the system of ODE

5.4.1 Blasius Solution in the Boundary Layer Theory

We present the basic idea of application of a CBP method to ODE based on Blasius boundary value problem.

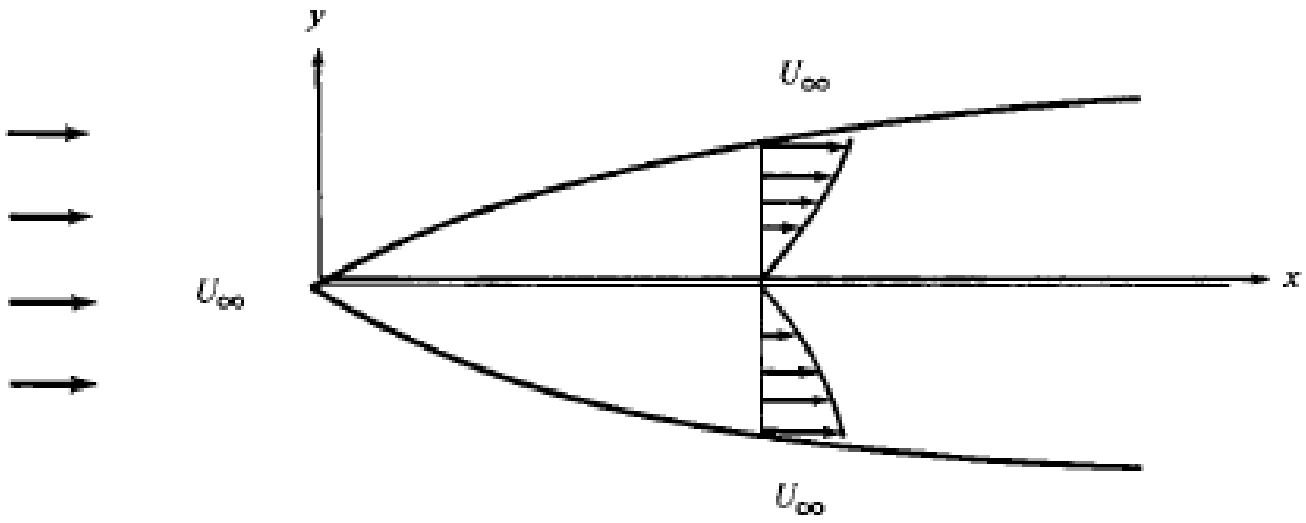


Figure 8: Boundary layer over the plate

Consider the flow of an incompressible fluid over a semi-infinite flat plate, as shown in Fig. 8. By introducing the boundary layer assumptions (i.e., the existence of a very thin layer), the Navier-Stokes equations becomes,

$$\left.\begin{aligned} &\frac{\partial u}{\partial x}+\frac{\partial v}{\partial x}=0 \\ &u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=v\frac{\partial^2 u}{\partial y^2} \end{aligned}\right\} \quad (1)$$

subject to the boundary conditions

$$\left.\begin{array}{ll} y = 0; & u = v = 0 \\ y = \infty; & u = U_\infty \end{array}\right\} \quad (2)$$

Blasius introduced the following transformation

$$u = \frac{\partial \Psi}{\partial y}, \quad v = -\frac{\partial \Psi}{\partial x} \quad (3)$$

$$\left.\begin{array}{l} \eta = y\sqrt{\frac{U_\infty}{\nu x}} \\ f(\eta) = \Psi/\sqrt{\nu x U_\infty} \end{array}\right\} \quad (4)$$

which transform the N-S equations to,

$$\frac{d^3 f}{d\eta^3} + \frac{1}{2} f \frac{d^2 f}{d\eta^2} = 0 \quad (5)$$

subject to the boundary conditions

$$f(0) = \frac{df(0)}{d\eta} = 0, \qquad \frac{df(\infty)}{d\eta} = 1$$

The following algorithm is presented as a sequence of steps:

Step 1: Present BVP in a canonical form as a system of the 1st order ODE

Introducing functions: $f'' = F_3; \quad f' = F_2; \quad f = F_1,$ rewrite BVP in the form

$$F_3' = -0.5F_1F_3, \quad F_1(0) = 0 \qquad (6)$$

$$F_2' = F_3, \qquad F_2(0) = 0 \qquad (7)$$

$$F_1' = F_2; \qquad F_2(\infty) = 1 \qquad (8)$$

Step 2: Introduce parameter p, and imbed obtained ODE in a p –parametric family

$$F_3' = -0.5[(F_1 - 1)p + 1]F_3, \quad F_1(0) = 0 \qquad (9)$$

$$F_2' = F_3, \qquad F_2(0) = 0 \qquad (10)$$

$$F_1' = F_2; \qquad F_2(\infty) = 1 \qquad (11)$$

Step 3: Obtain initial conditions at $p = 0$,

$$F_3 = 0.5e^{-\eta/2} \qquad (12)$$

$$F_2 = 1 - e^{-\eta/2} \qquad (13)$$

$$F_1 = \eta - 2(1 - e^{-\frac{\eta}{2}}) \qquad (14)$$

Step 4: Differentiating by parameter 'p', arrive at the following system with respect to sensitivities to the parameter p,

$$V' = AV + R, \qquad (15)$$

Where

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.5pF_3 & 0 & -0.5(pF_3 + 1) \end{bmatrix}; \quad R = \begin{bmatrix} 0 \\ 0 \\ -0.5F_3(F_1 - 1) \end{bmatrix} \qquad (16)$$

$$V = \frac{dF}{d\tau}; \qquad F = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \tag{17}$$

Step 5: Apply superposition principle and specify Cauchy problem for each component

$$V = aU + W \tag{18}$$

Where *U,W* – unknown vector functions; a – unknown "blend" coefficient

Solving the following two Cauchy problems for each component

$$U' = AU; \qquad U(0) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{19}$$

$$W' = AW + R; \qquad W(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{20}$$

we satisfy then automatically to the original ODE

$$(aU + W)' = A(aU + W) + R \tag{21}$$

Or

$$(aU' - AU) + (W' - AW - R) = 0 \tag{22}$$

and left boundary conditions.

Step 6: Solution of the Cauchy problems

Numerical scheme used in this work, is an implicit scheme, presented in the following form (*I* – identity matrix)

$$\frac{U^{i+1} - U^i}{\Delta\eta} = AU^{i+1}; \quad or \quad (I - \Delta\eta A)U^{i+1} = U^i \tag{23}$$

$$\frac{W^{i+1}-W^{i}}{\Delta\eta} = AW^{i+1} + R; \quad or \quad (I - \Delta\eta A)W^{i+1} = W^{i} + \Delta\eta R \qquad (24)$$

wherefrom,

$$U^{i+1} = (I - \Delta\eta A)^{-1}U^{i} \qquad (25)$$

$$W^{i+1} = (I - \Delta\eta A)^{-1}(W^{i} + \Delta\eta R) \qquad (26)$$

Step 7: Satisfy to the right BC by choosing the corresponding "blend" coefficient

Since the original right BC is applied only for the function F_2, as F_2=1, solving ODE for sensitivities we need to apply V_2=0, which in matrix form looks as *(J=[0 1 0])*

$$J * V = 0, or\ J * (aU + W) = 0 \qquad (27)$$

wherefrom, $$a = -\frac{J*W}{J*U} \qquad (28)$$

Figure 9 presents distribution of axial velocity (non-dimensionalized) across the normal distance to the wall.

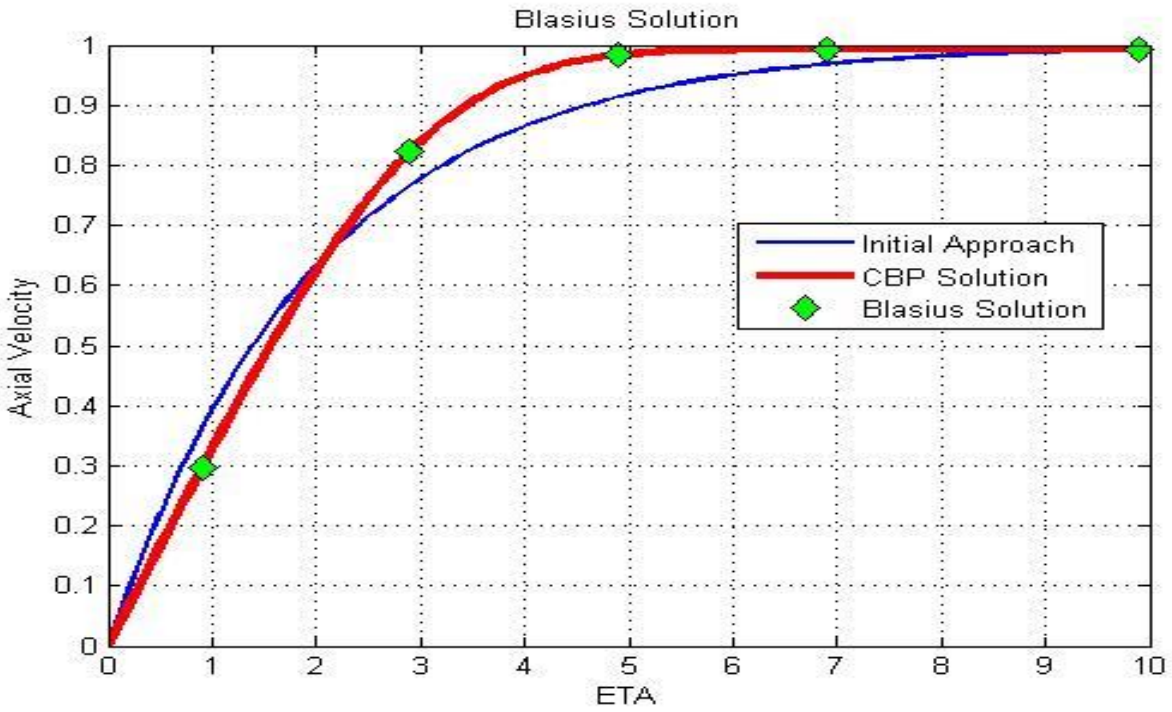


Figure 9: Velocity Profile of a Blasius Solution

5.4.2 Faulkner-Skan Solution of the Boundary Layer Flow over Wedges

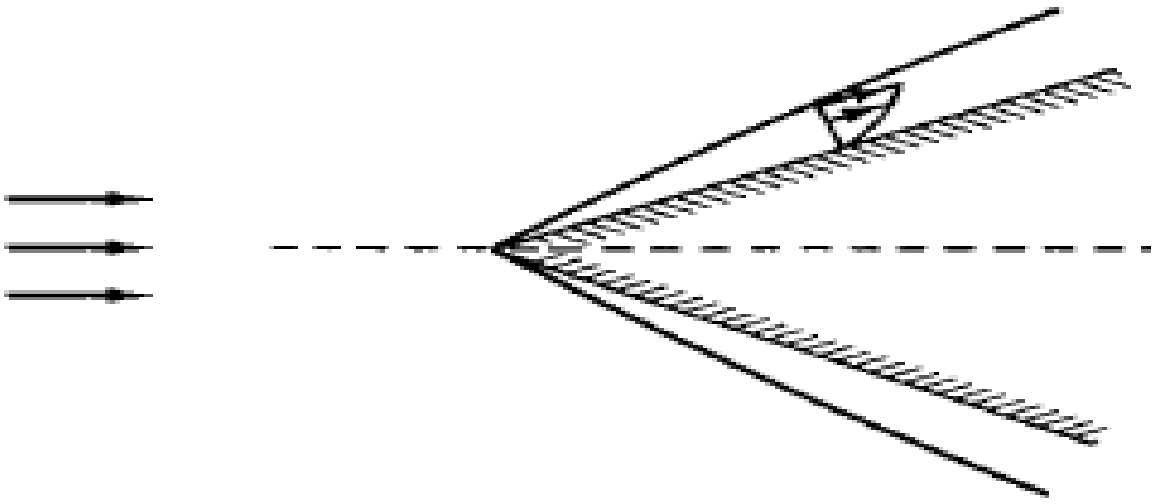


Figure 10: Boundary Layer Flow around the Wedge

We will now apply the method of parameter differentiation to the boundary layer equations governing the flow of fluids over the wedge as shown in Fig.10. The governing differential equations, derived from the Navier-Stokes equations, are identical to equations presented in section 5.4.1 except for an additional term, resulting from the fact that the mainstream velocity, i.e. the velocity at the edge of the boundary layer, is now a function of *x*. Similar to the Blasius equation, which derivation is based on a similarity transformation, the Falkner-Skan equation results in

$$\frac{d^e f}{d\eta^3} + f\frac{d^2 f}{d\eta^2} + \beta\left[1 - \left(\frac{df}{d\eta}\right)^2\right] = 0 \qquad (1)$$

subject to the boundary conditions

$$\left.\begin{array}{ll} \eta = 0; & f = 0,\ \frac{df}{d\eta} = 0 \\ \eta = \infty; & \frac{df}{d\eta} = 1 \end{array}\right\} \quad (2)$$

where 'β' relates to the power of a flow velocity profile. Solution of this equation has attracted the attention of both applied mathematicians and aeronautical engineers.

The mathematical algorithm of solving the Falkner-Skan equation is very similar to the one used for the solution of a Blasius boundary layer model, and is not presented here. Fig. 11 presents distribution of an axial velocity (non-dimensioned) as a function of a normal distance to the wall for different β. Distributions are in an excellent agreement with results presented in [37]. The corresponding MATLAB based text code is presented in Appendix.

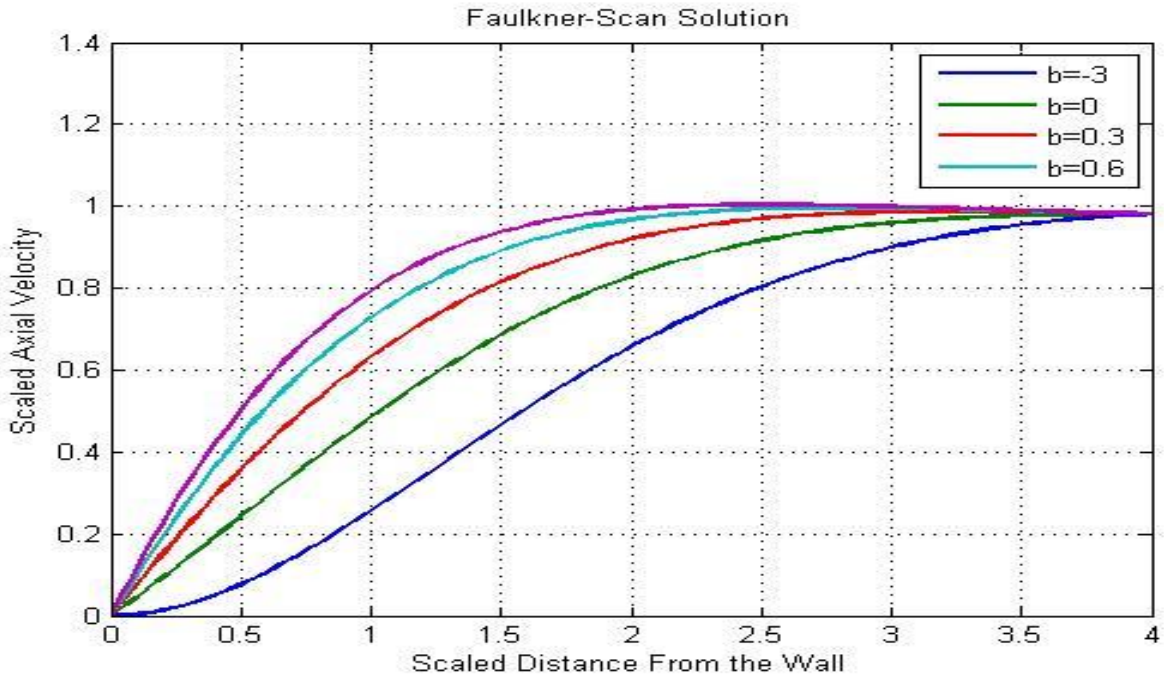


Figure 11: Velocity Profile across the Distance to the Wall for Faulkner-Skan solution

5.4.3 Rotating Disc Boundary Layer Flow

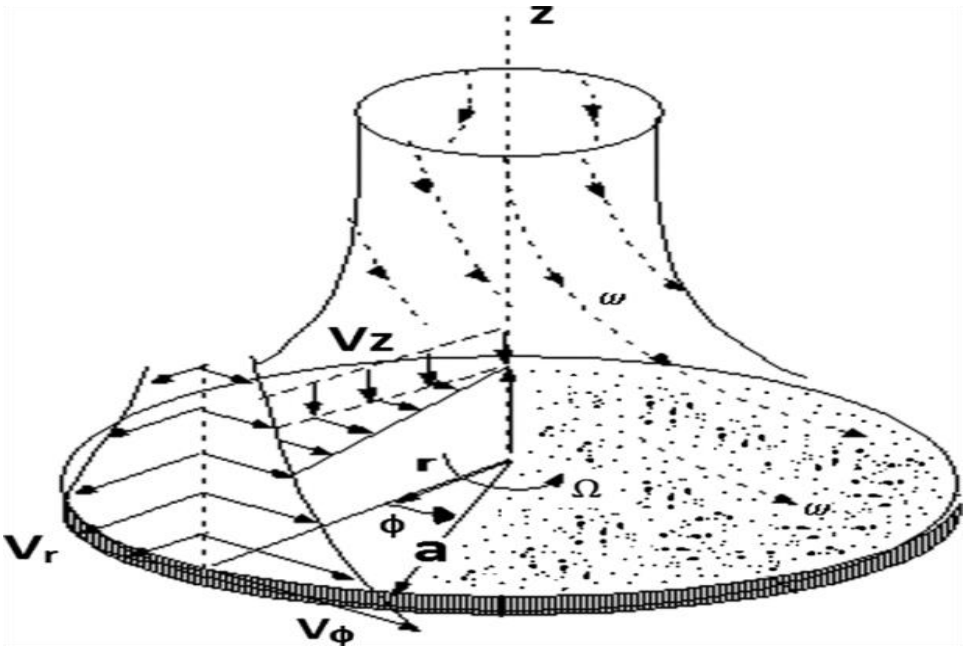


Figure 12: Flow over a rotating disc

The flow due to the rotating disc in a viscous fluid, as shown in the figure 12, was originally solved by Von Karman [32]. The original problem definition is available in Schlichting [32]. A disk of radius R is rotating with an angular velocity ω in still fluid. The flow is steady, incompressible, has constant property, and is axisymmetric. The fluid at the disk has to satisfy the no slip condition. The centrifugal effects cause the fluid to leave the disk radially near the disk. The flow above the disk must replace this airflow through a downward spiraling flow. A cylindrical coordinate system (r, θ, z) is used for description. Vr, Vθ, Vz, are the velocity components, p is the pressure, ν, the kinematic viscosity. Von Karman showed that the Navier-Stokes equations can be solved by casting them into a non-dimensional form by using certain substitutions. This leads to a set of three coupled non-linear ordinary differential equations for

the dimensionless functions F, G and H, and an equation for the dimensionless dynamic pressure P in the fluid above the disk [33]:

$$\left.\begin{aligned} 2F + \dot{H} &= 0 \\ F^2 + \dot{F}H - G^2 - \ddot{F} &= 0 \\ 2FG + H\dot{G} - \ddot{G} &= 0 \\ \dot{P} + H\dot{H} - \ddot{H} &= 0 \end{aligned}\right\} \quad (1)$$

The following boundary conditions supplement the ODE system (1),

$$\left.\begin{aligned} \eta = 0\,; F = 0; G = 1; H = 0; P = 0 \\ \eta = \infty\,; F = 0; G = 0 \end{aligned}\right\} \quad (2)$$

The mathematical algorithm of solving these equations is very similar to the one used for the solution of a Blasius boundary layer model and the Faulknar-Skan's equation. Table 1 shows the comparison between the published results by Sparrow and Gregg with the CBP method for a coarse and fine mesh. It is evident that the CBP solution for coarse mesh provided results closer to publish values with very less time. Whereas, the fine mesh provides solution approximately equal to the published results with a comparatively more solution time. Hence, we can say CBP provides better results for the finest mesh with better time for convergence.

	E.M. Sparrow and J.J Gregg values			CBP method values (Coarse Mesh-100 cells)			CBP method values (Fine Mesh-3500 cells)		
η	$\dot{F}$	$-\dot{G}$	$-\dot{H}$	$\dot{F}$	$-\dot{G}$	$-\dot{H}$	$\dot{F}$	$-\dot{G}$	$-\dot{H}$
0	0.510	0.6159	0	0.5284	0.6216	0	0.5133	0.6189	0
∞	0	0	0.8845	0.0001	0.0001	0.8644	0.0001	0.0001	0.8859
Time for convergence of solution				2.3 seconds			65 seconds		

Table 1: Shows the comparison of the solution values obtained by E.M. Sparrow and J.J. Gregg, and the values obtained by the proposed CBP method.

The graphical representations of these results are shown below. Results were compared with the published results obtained by Schlichting and the results obtained using Bezier functions to solve the differential equation with the given boundary conditions [32]. Figure 13 shows the comparison between CBP method results and results presented by Schlichting [32].

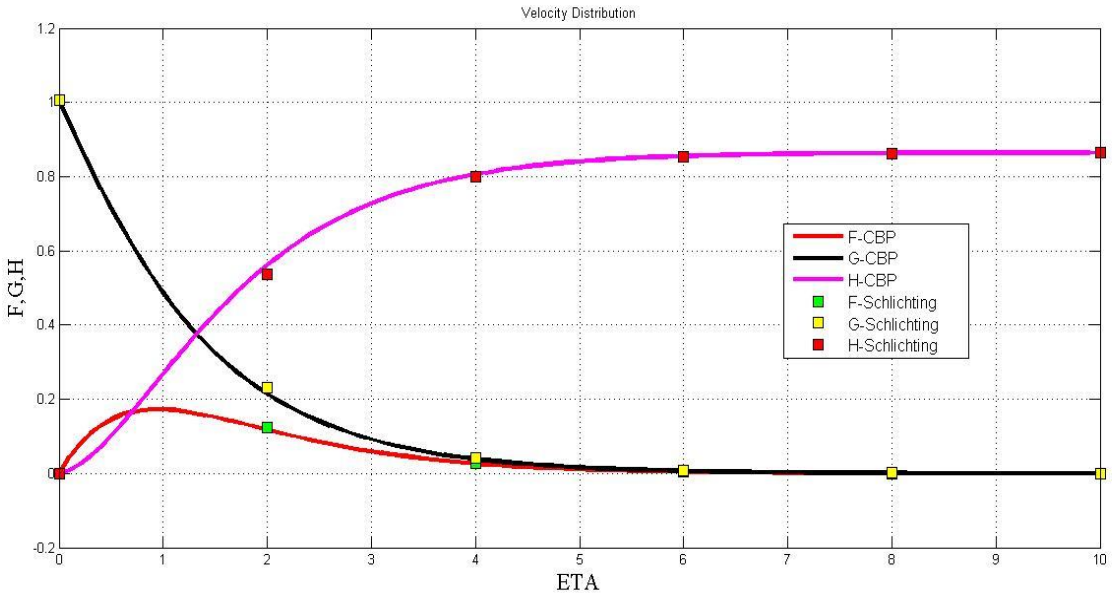


Figure 13: The plot shows the comparison for velocity distribution between the results obtained by the proposed method and the results obtained from Schlichting

5.4.4 General Solution for Troesch's Problem

Troesch' problem is an excellent example for testing different numerical schemes. Originally derived to predict the confinement of a plasma column, it serves as a difficult testing case for different algorithms. The continuation by parameter method is employed to obtain the solution for the nonlinear differential equation which describes Troesch's problem. In contrast to other reported solutions obtained by various methods, the proposed solution shows the highest degree of accuracy in the results for a remarkable wide range of values of Troesch's parameter.

Troesch's boundary value problem, in terms of dimensionless variables can be written as:

$$\frac{d^2y}{dx^2} = n \sinh ny \qquad (1)$$

subject to the boundary conditions

$$y(0) = 0, \qquad y(1) = 1 \qquad (2)$$

Following the described procedure relating to the CBP method, we present equation (1) in a form of a system of two nonlinear equations of the first order.

$$y' = z$$

$$z' = n \sinh ny$$

Differentiating with parameter 'n' results in the following "sensitivity" equations

$$\bar{y} = \frac{dy}{dn}; \quad \bar{z} = \frac{dz}{dn}$$

$$\bar{y}' = \bar{z}$$

$$\bar{z}' = \sinh ny + n \cosh ny \, (y + n\bar{y})$$

being supplemented by the boundary conditions,

$$\bar{y}(0) = 0;\ \bar{y}(1) = 0$$

Following stepwise procedure and updating the solution for incrementally increased parameter 'n' provides a single step solutions which looks as,

$$y(n + dn) = y(n) + \bar{y}\Delta n$$

$$z(n + dn) = z(n) + \bar{z}\Delta n$$

In matrix form, $\bar{V}' = A\bar{V} + F$

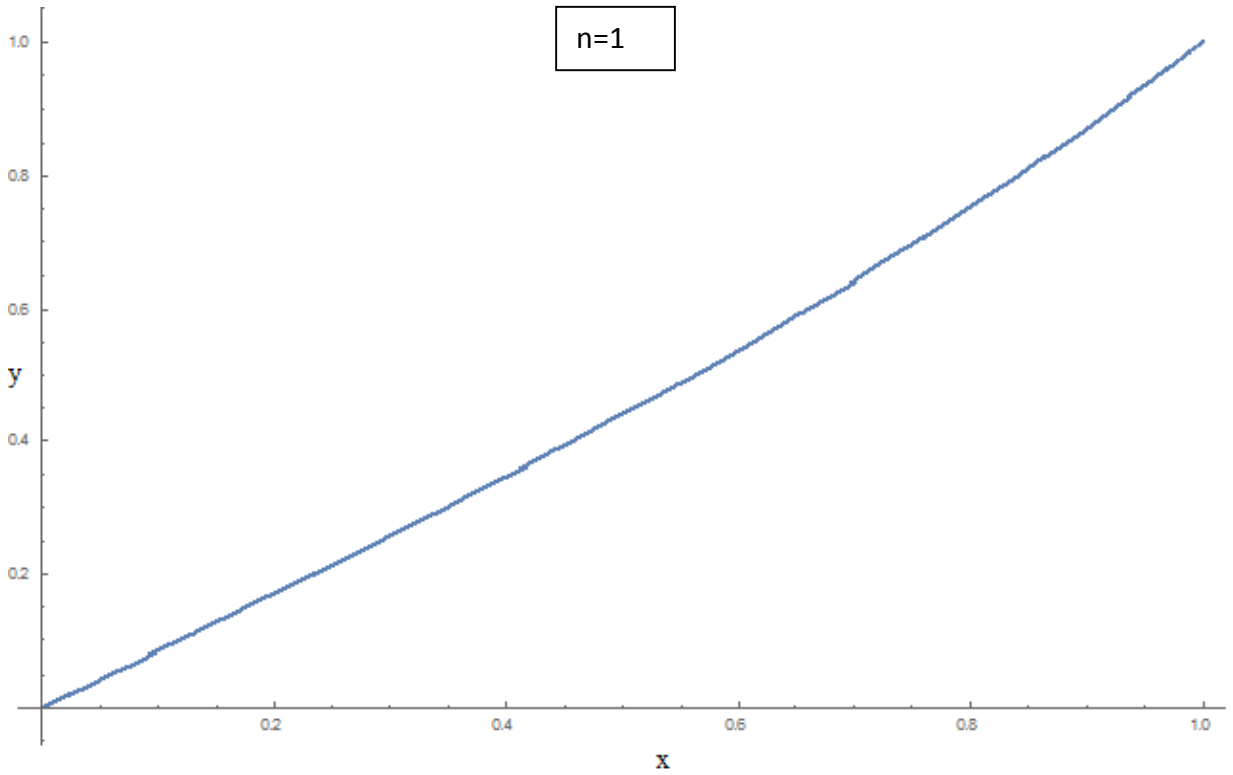
$$V = \begin{pmatrix} y \\ z \end{pmatrix} \qquad \bar{V} = \begin{pmatrix} \bar{y} \\ \bar{z} \end{pmatrix}$$

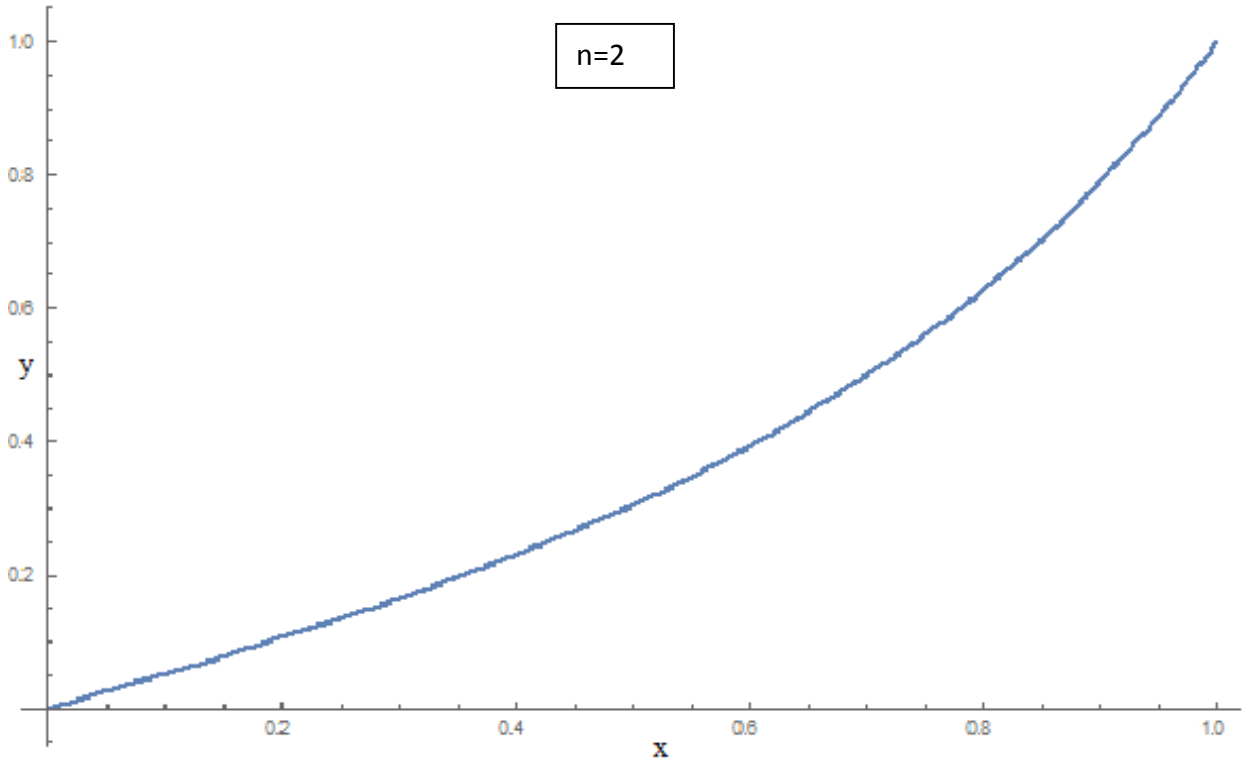
$$A = \begin{pmatrix} 0 & 1 \\ n^2 \cosh ny & 0 \end{pmatrix}; \qquad F = \begin{pmatrix} 0 \\ \sinh ny + ny \cosh ny \end{pmatrix}$$

Furthermore,

$$V(n + \Delta n) = V(n) + \bar{V}\Delta n \qquad (3)$$

Easy to see that the large number for 'n' correspond to the steep edge effect at the right boundary condition for the presented nonlinear BVP. Troesch [34] and Ehrlich [35] pointed out that this two-point boundary value problem is unstable and difficult to solve. Roberts and Shipman [36] suggested that the equation can be solved by combination of three different methods, namely, the perturbation technique, the parallel shooting method, and the continuation method. The combination of these methods is necessary since none of these methods by itself is sufficient. However, this method gives accurate solutions of the Eq. (1) for n < 5. For $n \geq 5$, their results do not converge.


n=1
1.0
0.8
0.6
y
0.4
0.2
0.2
0.4
0.6
0.8
1.0
x


n=2
1.0
0.8
0.6
y
0.4
0.2
0.2
0.4
0.6
0.8
1.0
x

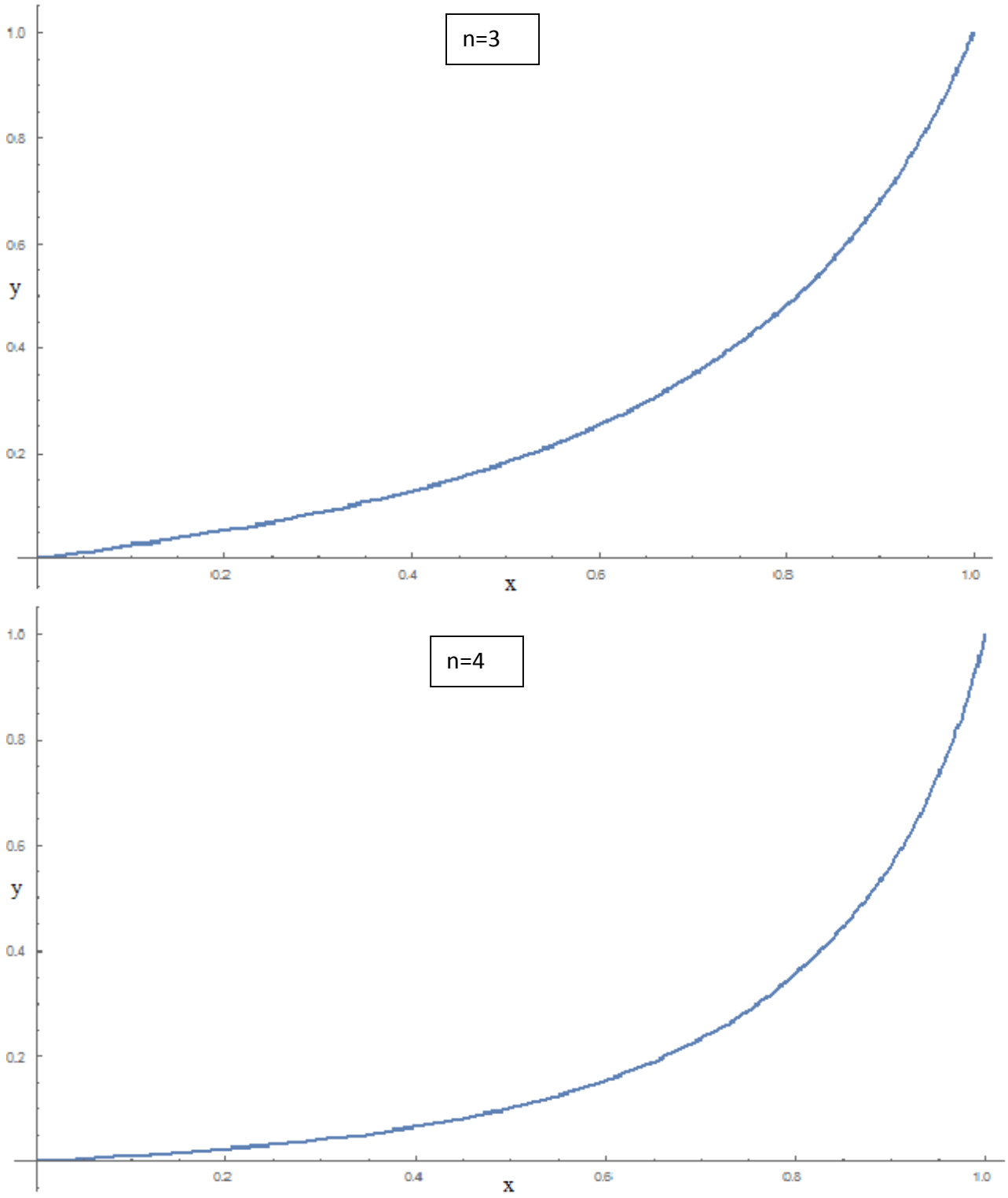
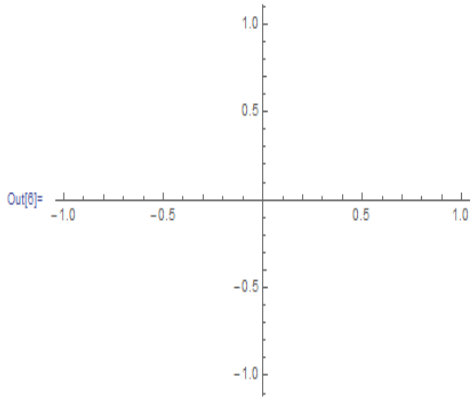


Figure 14: The figure shows the solution of Troesch's equation using Wolfram Mathematica for n=1, n=2, n=3 and n=4 respectively

Now to compare the results, the Troesch's equation was solved using commercial computing software Wolfram Mathematica. Mathematica has a built-in command solver called 'NDSolve' which can be used to solve this problem. As it can be seen from the figure above, the result obtained for lower parametric value 'n' is stable. But when the value of the parameter exceeds 4 ($n > 4$), solution becomes unstable and diverges.

```
In[5]:= s = NDSolve[{y''[x] - 5 Sinh[5 y[x]] == 0, y[0] == 0, y[1] == 1}, y[x], {x, 0, 1}] // Simplify
NDSolve::ndsz : At x == 0.9555766885690306`, step size is effectively zero; singularity or stiff system suspected. >>
Out[5]= NDSolve[{5 Sinh[5 y[x]] == y''[x], y[0] == 0, y[1] == 1}, y[x], {x, 0, 1}]

In[6]:=
Plot[Evaluate[y[x] /. s], {x, 0, 1}, PlotRange → All]
ReplaceAll::reps : {NDSolve[{5 Sinh[5 y[<<1>>]] == y''[x], y[0] == 0, y[1] == 1}, y[x], {x, 0, 1}]} is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. >>
NDSolve::dsvar : 0.000020428571428571424` cannot be used as a variable. >>
ReplaceAll::reps : {NDSolve[{5 Sinh[5 y[<<1>>]] == y''[0.0000204286], y[0] == 0, y[1] == 1}, y[0.0000204286], {0.0000204286, 0, 1}]} is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. >>
NDSolve::dsvar : 0.000020428571428571424` cannot be used as a variable. >>
ReplaceAll::reps : {NDSolve[{5. Sinh[5. y[<<1>>]] == y''[0.0000204286], y[0.] == 0., y[1.] == 1.}, y[0.0000204286], {0.0000204286, 0., 1.}]} is neither a list of replacement rules nor a valid dispatch table, and so cannot be used for replacing. >>
General::stop : Further output of ReplaceAll::reps will be suppressed during this calculation. >>
NDSolve::dsvar : 0.02042859183673469` cannot be used as a variable. >>
General::stop : Further output of NDSolve::dsvar will be suppressed during this calculation. >>
```



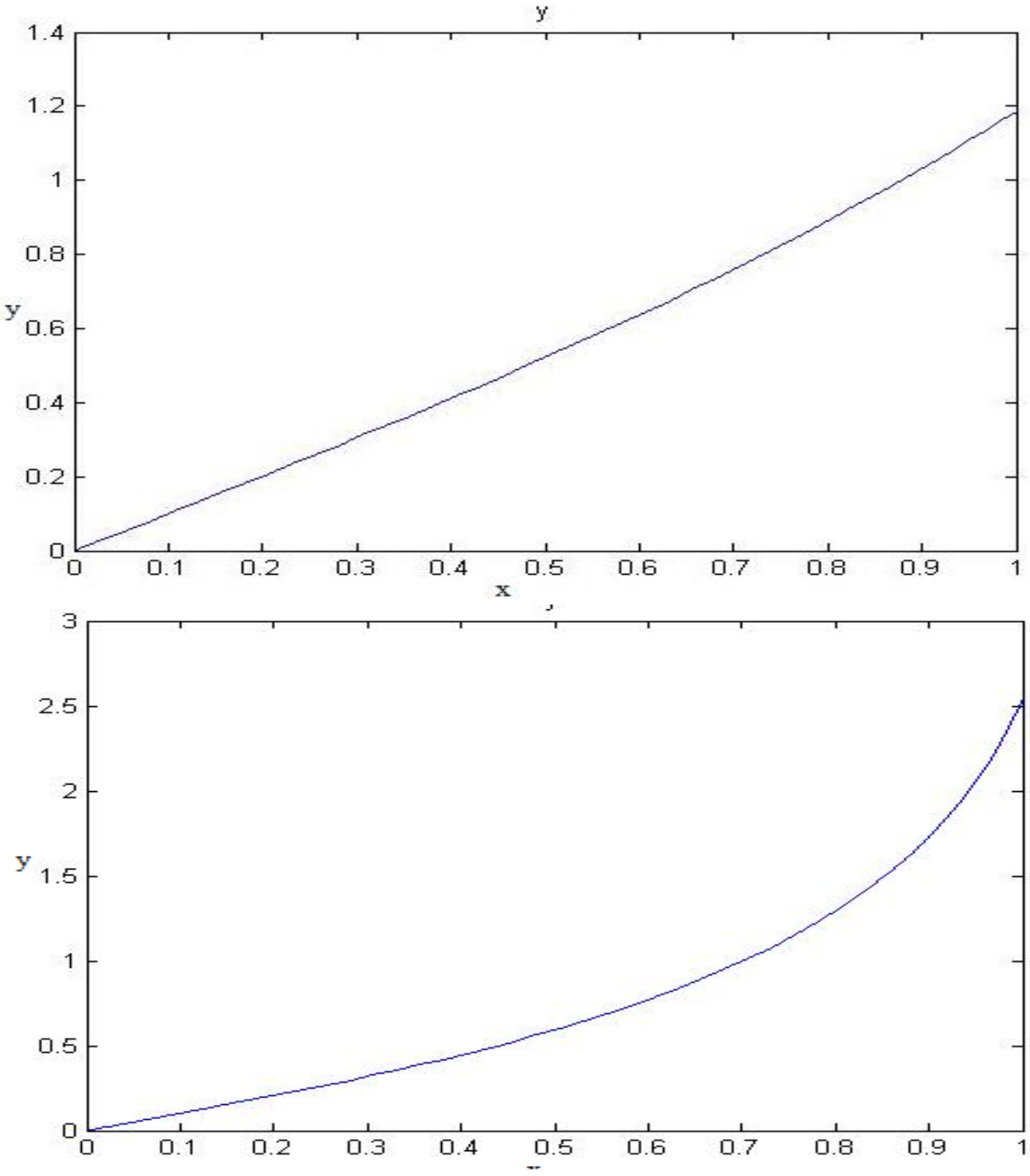


Figure 15: The figure shows the solution of Troesch's equation using MatLab for n=1 and n=2 respectively

```
>> Troeschcode
Warning: Failure at t=7.188468e-01.  Unable to meet integration tolerances without reducing
the step size below the smallest value allowed (1.776357e-15) at time t.
> In ode45 at 309
  In Troeschcode at 4
```

The MATLAB's ODE toolbox was also used to solve the Troesch's equation. As compared to the solution obtained by Mathematica, the solution obtained using MATLAB was stable for lower parametric values but as the parametric values goes above 2, the program crashes and MATLAB is unable to provide any results. We can see that it is really difficult to obtain a solution for large parametric values.

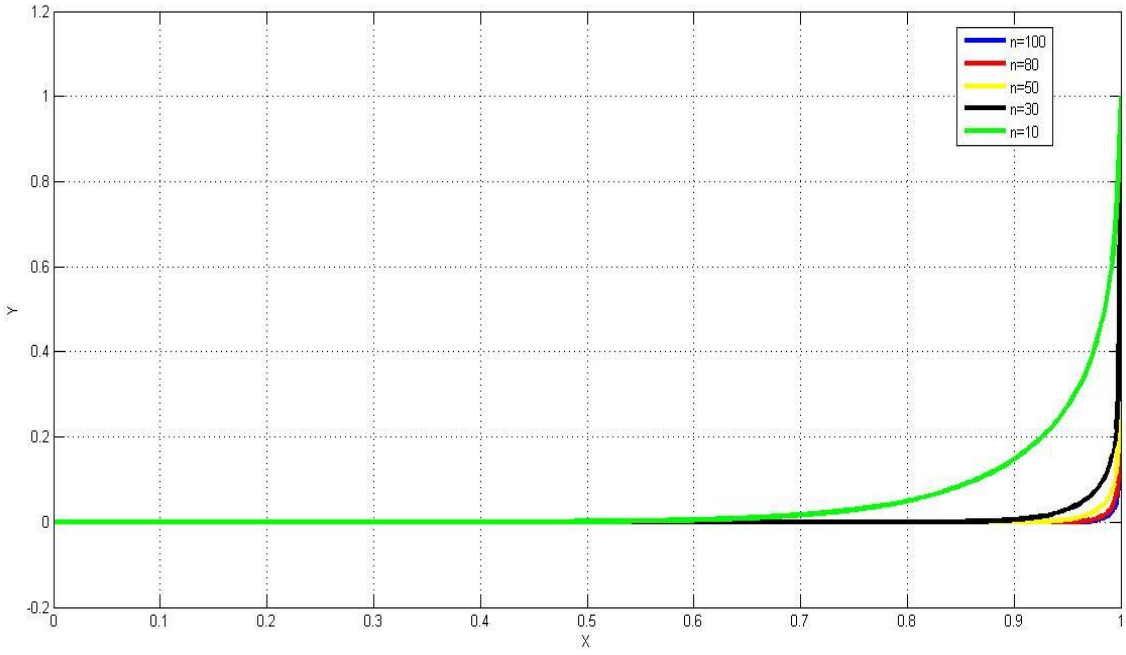


Figure 16: The figure shows the solution of Troesch's equation using continuation by parameter method for varying n.

An efficient algorithm based on the continuation by parameter method has been successfully applied to Troesch's problem which provides the stable solutions for higher values of parameter 'n'. As it can be seen from the figure 17, the continuation by parameter provides us a stable solution for a parametric value up to n=100. It is important to emphasize that for the first time in the literature of the Troesch's problem, the solution is plotted for large values of n=100.

5.5 An Alternative Formulation of Bellman's Method of Invariant Imbedding

The classical invariant imbedding is based on a partitioning of an interval of integration, with a following integration across each partition [3], [4], [5], [6].

We suggest a modification based on an introduction of a parameter of continuation, which is a range of integration of a relating BVP, varying from zero to the nominal value

$$\frac{dY}{dx} = F(x, Y); \;\; x = \bar{x}l; \qquad \hat{Y} = \frac{dY}{dl}$$ - sensitivities to the length change

$$\frac{dY}{d\bar{x}} = lF(x, Y);$$ l=0 – corresponds to initial approach

Linear equation for the vector of sensitivities $\frac{d\hat{Y}}{d\bar{x}} - lF_Y\hat{Y} = F + lF_x\bar{x}$

Solution update

$$Y = Y + \hat{Y}\Delta l$$

Discrete counterpart can be suggested as well. Approximating original ODE by finite difference equation of a second order of accuracy, obtain

$$\frac{y_{i-1} - 2y_i + y_{i+1}}{L^2} N^2 = n\sinh(ny_i)$$

Or

$$y_{i-1} - 2y_i + y_{i+1} = \frac{l}{N^2}\, n\sinh(ny_i), \qquad l = L^2$$

Introducing derivatives by l, $\frac{dy}{dl} = \bar{y}$, rewrite basic equations in terms of sensitivities to the span variation

$$\overline{y_{i-1}} - 2\overline{y_i} + \overline{y_{i+1}} = \frac{l}{N^2}\, n[\sinh(ny_i) + lcosh(ny_i)n\bar{y}_i],$$

The final tri-diagonal system that is one with a bandwidth of 3 can be expressed as

$$a\bar{y}_{i-1} + b\bar{y}_i + c\bar{y}_{i+1} = r_i$$

$$a = c = 1; \;\; b = -2 - \left(\frac{n}{N}\right)^2 lcosh(ny_i); \;\; r = \frac{n}{N^2} sinh(ny_i)$$

Solution of obtained tridiagonal equations is obtained by the Thomas method. Updated solution is given as,

$$y_i = y_i + \bar{y}_i \Delta l, \text{ i=1,…, N.}$$

At initial point, when span is equal to zero, solution is a linear interpolation of boundary values inside domain, i.e. $y_i = \frac{i-1}{N-1}, i = 1, \dots, N$

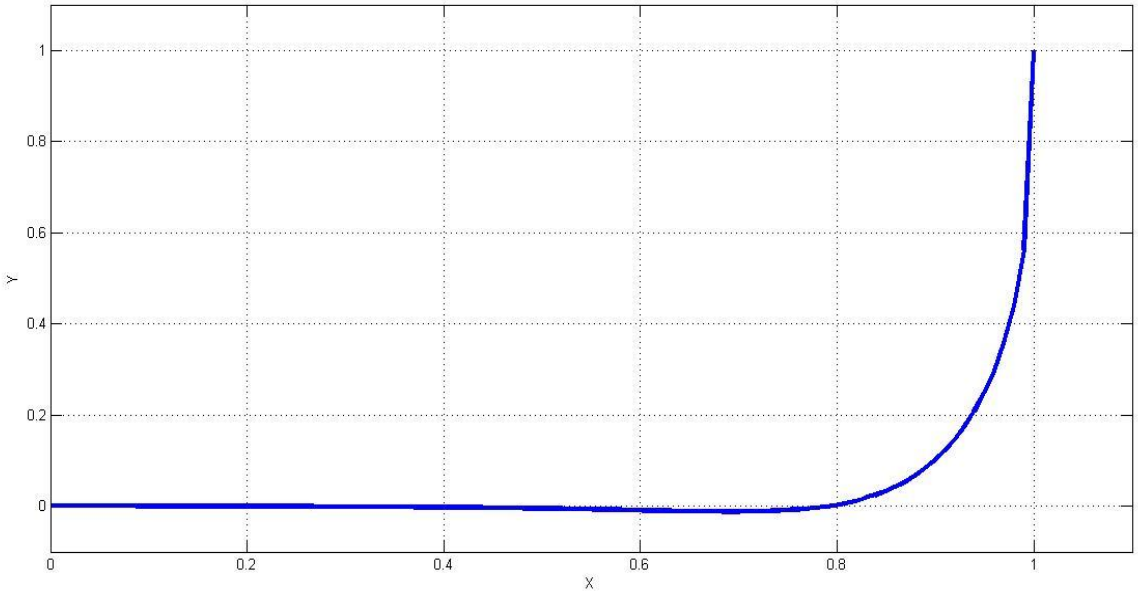


Figure 17: The figure shows the solution of Troesch's equation using invariant imbedding method for n=12.

6. CONCLUSION

Continuation by parameter method has been applied to different nonlinear boundary value problems from different areas of engineering like fluid mechanics, mechanics of solid, stability, etc.

It was found that modifications of CBP method solved successfully all the boundary value problems including notoriously famous ‘stiff’ Troesch’s problem. This method provided solution within the range of the Troesch’s parameter, exceeding much the ones appeared to be critical for the solvers used by other methods, as well as the currently available commercial software like Wolfram Mathematica and Matlab.

7. REFERENCES

[1]. Ambarzumian, V. A., "Theoretical Astrophysics," Pergamon, Oxford, 1958.

[2]. Chandrasekhar, S., "Radiative Transfer," Dover, New York, 1960.

[3]. Arosesty, J., Bellman, R., Kalaba, R., and Ueno, S., Invariant imbedding and rarefied gas dynamics, *Proc. Natl. Acad. Sci.* 50,222 (1963).

[4]. Bellman, R., Kalaba, R., and Prestrud, M. C., "Invariant Imbedding and Radiative Transfer in Slabs of Finite Thickness," American Elsevier, New York, 1963.

[5]. Bellman, R., Kagiwada, H. H., Kalaba, R. E., and Prestrud, M. c., "Invariant Imbedding and Time-Dependent Transport Processes," American Elsevier, New York, 1964.

[6]. Bellman, R., and Kalaba, R. E., Invariant imbedding, random walk, and scattering, *J. Math. Mech.* 9, 411 (1960).

[7]. Bellman, R., and Kalaba, R. E., Wave branching processes and invariant imbedding, *Proc. Natl. Acad. Sci.* 47, 1507 (1961).

[8]. Bellman, R., and Kalaba, R. E., A Note on Hamilton's equations and invariant imbedding, *Q. Appl. Math.* 21, 166 (1963).

[9]. Wing, G. M., "An Introduction to Transport Theory," Wiley, New York, 1962.

[10]. Lee, E. S., "Quasilinearization and Invariant Imbedding," Academic Press, New York, 1968.

[11]. Meyer, G. H., "Initial Value Methods for Boundary Value Problems," Academic Press, New York, 1973.

[12]. Scott, M. R., "Invariant Imbedding and Its Applications to Ordinary Differential

Equations: An Introduction," Addison-Wesley, Reading, Massachusetts, 1973.

[13]. Na T.Ya. Computational methods in boundary value problems. Mathematics in science and engineering. Volume 145. 1979

[14]. Grigolyuk E.I.,Shalashilin V.I. Problems of nonlinear deformation. 1991

[15]. Ortega, J.M.,Rheinboldt, W.C., "Itertaive Solution of Nonlinear Equations in Several Variables", Academic Press. New Ypork – London, 1970

[16]. Bittner, L. ISNM, International Series of Numerical Mathematics, Vol. 7, Verlag, Basel, SHttgart (1987), pp.114-135

[17]. Distefano, N., Todeschini, R. A quasolinearization approach to the solution of elastic beams on nonlinear foundations, Int. J. Solids Struct, 11,No 1, 89-97, 1975

[18]. El-Zanaty, M.H., Murray D.W. Non linear finite element analysis of steel frames, J.Struct Eng., 109, 2, 353-368, 1983

[19]. Vorovich I.I.,Zipalova, V.F. Solution of nonlinear boundary value problems by passing to a Cauchy problem. Applied Mathematics and Mechanics., 29, N4, 357-367, 1965

[20].On the numerical solution of snapping problems in the theory of elastic stability. Sudaar #410, Stanford, CA, 1982

[21]. Ricks, An incremental approach to the solution of snapping and buckling problems. Int. J. Solids Structures 15,524-551 (1979).

[22]. Ricks, E. The application of Newton's method to the problem of elasticity, Trans. ASME J.Appl Mech, E39, No4, 1060-1065, 1972

[23]. Rothert, H, et all Snap through buckling of reticulated space trusses. J.Struct. Div. Proc, ASCE, 107, No1, 129-143 (1981)

[24]. Waszczyszyn, Z Numerical problems of nonlinear stability analysis. Comput. Structures, 17, No 1, 69-72, 1983

[25]. Watson , L.T., Holzer, S.M. Quadratic convergence of Crisfield's method. Comput. Structures, 17. No1, 69-72, 1983

[26]. Walker A.S., Hall D.G. AN analysis of the large deflection of beams using finite element method. Aeronaut. Quart., 19. No4, 357-367, 1968

[27]. Weinitschke H.J., On the calculation of limit and bifurvation points in stability problems of elastic shells. Int, J. Solid. Struct, 24, No1, 79-85, 1985

[28]. Valli, A. M., Elias, R. N., Carey, G. F., & Coutinho, A. L. (2009). PID adaptive control of incremental and arclength continuation in nonlinear applications. International Journal for Numerical Methods in Fluids Int. J. Numer. Meth. Fluids, 61(11), 1181-1200. doi:10.1002/fld.1998

[29]. Keller H.B. Numerical solution of bifurcation and nonlinear eigenvalue problems. In P.H.Rabinowitz, Application of bifurcation theory, 359-384, Academic Press, NY, 1977

[30]. Stability and oscillation of elastic systems: Modern concepts, paradoxes and errors : Panovko, Y. G. : Free Download & Streaming : Internet Archive. (n.d.). Retrieved March 10, 2016, from https://archive.org/details/nasa_techdoc_19740006513

[31]. Miklavcic, M., & Wang, C. Y. (2004). The flow due to a rough rotating disk. Zeitschrift For Angewandte Mathematik Und Physik (ZAMP), 55(2), 235-246.

[32]. Schlichting, H. (1979). Boundary-layer theory. Exact solution of Navier-Stokes equations, pg 102.

[33]. Venkataraman, P. (n.d.). Explicit Solutions for Differential equations. Retrieved March 10, 2016, from https://people.rit.edu/~pnveme/ExplictSolutions2/ Explicit Solutions to Differential Equations, Flow over a Rotating Disk

[34]. B.A. Troesch, Intrinsic difficulties in the numerical solution of a boundary value problem, Internal Report NN–142, TRW Inc., Redondo Beach, California, 1960

[35]. L. EHRLICH, Experience with Numerical Methods for a Boundary Value Problem, Internal Report NN-141, TRW Inc., Redondo Beach, Calif., 1960

[36]. S.M. Roberts, J.S. Shipman, Solution of Troesch's two-point boundary value problem by a combination of techniques, J. Comput. Phys., 10 (1972), pp. 232–241

[37]. Nandy, Samir Kumar, and Rajib Kumar Mandal. "Public Science Framework-Journals - Paper - HTML." *MHD Stagnation-Point Flow and Heat Transfer of Nanofluid over a Shrinking Surface*. Journal of Nanoscience and Nanoengineering, Vol. 1, No. 4, 13 Sept. 2015. Web. 04 Apr. 2016.

A. APPENDIX

MATLAB Computer Text Codes

Bernoulli Lemniscate

```
clc; clear all; close all
%Plot Bernoulli Lemniscate
x1=sqrt(2); x2=0;
AX1(1)=x1; AX2(1)=x2;  tau=0.00001; a=1;

for i=2:1000000
    J1=(x1^2+x2^2)*x1-a^2*x1;
    J2=(x1^2+x2^2)*x2+a^2*x2;
    J=sqrt(J1^2+J2^2);
    x1=x1+tau*J2/J; x2=x2-tau*J1/J;
    AX1(i)=x1; AX2(i)=x2;
end
%Filter points for plotting
k=0;
for i=1:10000:1000000
    k=k+1;
    X(k)=AX1(i); Y(k)=AX2(i);
end
% plot(AX1,AX2); grid on; xlabel('X1'); ylabel('X2');
% hold on
```

```
21. plot(X,Y,'rs','MarkerEdgeColor','k',...
22.     'MarkerFaceColor','g',...
23.     'MarkerSize',10)
24. %Exact solution
25. t=linspace(0,2*pi);
26. X=a*sqrt(2)*cos(t)./(1+sin(t).^2);
27. Y=a*sqrt(2)/2*sin(2*t)./(1+sin(t).^2);
28. hold on
29. plot(X,Y);
30. grid on; xlabel('X1'); ylabel('X2');
31. legend('CBP Method','Exact Solution')
32.
33. %Newton=-Raphson
34.
35. x=-sqrt(2); y=1;
36. for ii=1:101
37.     for k=1:100
38.         F=(x^2+y^2)^2-2*a^2*(x^2-y^2);
39.         DFY=4*((x^2+y^2)*y+a^2*y);
40.         DY=F/DFY;
41.         y=y-DY;
42.         if(abs(DY)<0.0001)
43.             DY,k
```

```
44.            break
45.        end                ;
46.     end %k
47.     XN(ii)=x; YN(ii)=y;
48.     ii,x=x+2*sqrt(2)/100
49. end %ii
50. figure
51. plot(XN,YN,'rs','MarkerEdgeColor','k',...
52.     'MarkerFaceColor','g',...
53.     'MarkerSize',10);
54. hold on
55. plot(X,Y);
56. grid on; xlabel('X1'); ylabel('X2');
57.
58. legend('Newton-Raphson','Exact Solution')
```

Gas mixture composition

```
%Equilibrium Mixture

clc; close all; clear all


Ntau=100; %number of intervals

dtau=1/Ntau;

TAU=0:dtau:1;


al=0.5; bt=0.5; gm=0.5;

AL(1)=al; BT(1)=bt; GM(1)=gm; K=exp(0.61); p=10;

V=[al bt gm ]; %Initial vector at tau=0


A(1,1)=1; A(1,2)=1; A(1,3)=0;

A(2,1)=2; A(2,2)=1; A(2,3)=2;


for i=1:Ntau

   F(1)=-(al+bt-1);

   F(2)=-(2*al+bt+2*gm-2);

   F(3)=K^2*al^2*(al+bt+gm)-bt^2*gm*p;


   A(3,1)=-K^2*(3*al^2+2*al*bt+2*al*gm);

   A(3,2)=2*bt*gm*p-K^2*al^2

   A(3,3)=bt^2*p-K^2*al^2;
```

```
    DV=A\F';
    V=V+DV'*dtau;
    %Updated
    al=V(1); bt=V(2); gm=V(3);
    AL(i+1)=al; BT(i+1)=bt; GM(i+1)=gm;
end
V

plot(AL,'--rs','LineWidth',2,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','r',...
                'MarkerSize',10)
hold on
plot(BT,'--bd','LineWidth',2,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','b',...
                'MarkerSize',10)
hold on
plot(GM,'--cd','LineWidth',2,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',10)
```

```
            title('Mole Coefficients. Convergence ')
grid on
legend('Alpha','Betta','Gamma')
xlabel('Iterations')
ylabel('Mole Numbers of Components')
title('Convergence')
```

Blasius Boundary Layer Model

```
% Solving Blasius BV problem:
%f'''+0.5*f*f''=0;  f(0)=f'(0)=0; f'(inf)=1;
clc; close all; clear all
N=100; %number of layers from eta=0 to eta=10 (inf)
Ntau=100; %number of tau imcrements
dtau=1/Ntau;
deta=10/N;
%Exact solution of a simplified model at initial state (tau=0)
ETA=linspace(0,10,N+1);
F3=0.5*exp(-ETA/2);
F2=1-exp(-ETA/2);
F1=ETA-2*(1-exp(-ETA/2));
plot(ETA,F2,'LineWidth',2)
hold on
tau=0;
for itau=1:Ntau
    U=[0;0;1]; %initial conditions for sensitivities
    W=[0;0;0];
    AU(1,:)=U;  AW(1,:)=W;

    for i=1:N  %integrate eq for sensitivities U and W
```

```
        M=[0 1 0; 0 0 1; -tau*F3(i)/2 0 -(tau*(F1(i)-1)+1)/2];
        R=[0;0;-F3(i)*(F1(i)-1)/2];

        DINV=inv(eye(3)-deta*M);
        U=DINV*U;
        W=DINV*(W+deta*R);
        AU(i+1,:)=U;    AW(i+1,:)=W;
    end %i

    %Blend coefficient
    A=-W(2)/U(2);

    %Total vector of sensitivities
    AV=A*AU+AW;

    %Update F functions
    F1=F1+dtau*AV(:,1)';
    F2=F2+dtau*AV(:,2)';
    F3=F3+dtau*AV(:,3)';
    tau=tau+dtau;
end  %itau

plot(ETA,F2,'r','LineWidth',3)
```

```
grid on
xlabel('ETA'); ylabel('Axial Velocity'); title('Blasius Solution')


hold on
XX=[ETA(10), ETA(30), ETA(50), ETA(70), ETA(100)];
YY=[F2(10), F2(30), F2(50), F2(70), F2(100)];
plot(XX,YY,'bd',...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',10)
legend('Initial Approach', 'CBP Solution', 'Blasius Solution')
```

Falkner-Skan Flow across the edge

```
% Solving Falkner-Skan BV problem:


clc; close all; clear all
N=100; %number of layers from eta=0 to eta=10 (inf)
Ntau=100; %number of tau imcrements
dtau=1/Ntau;
deta=4/N;


b=-0.3;
b=0;
b=0.3;
b=0.6
for ib=1:5
   b=-0.3+(ib-1)*0.3;
% for b=-0.3:0.3:1.2 %-0.3:0.3:0.6


%Exact solution of a simplified model at initial state (tau=0)
ETA=linspace(0,4,N+1);
F3=exp(-ETA);
F2=1-exp(-ETA);
F1=-1+ETA+exp(-ETA        );
% plot(ETA,F2)
```

```
% hold on

tau=0;
for itau=1:Ntau
   U=[0;0;1]; %initial conditions for sensitivities
   W=[0;0;0];
   AU(1,:)=U;  AW(1,:)=W;

   for i=1:N  %integrate eq for sensitivities U and W

      M=[0 1 0; 0 0 1; -tau*F3(i) 2*tau*b*F2(i) -(tau*(F1(i)-1)+1)];
      R=[0;0;-F3(i)*(F1(i)-1)+b*(F2(i)^2-1)];

      DINV=inv(eye(3)-deta*M);
      U=DINV*U;
      W=DINV*(W+deta*R);
      AU(i+1,:)=U;    AW(i+1,:)=W;
   end %i

   %Blend coefficient
   A=-W(2)/U(2);

   %Total vector of sensitivities
```

```
    AV=A*AU+AW;


    %Update F functions
    F1=F1+dtau*AV(:,1)';
    F2=F2+dtau*AV(:,2)';
    F3=F3+dtau*AV(:,3)';
    AF2(:,ib)=F2;
end  %itau
% get(0,'DefaultAxesColorOrder');
% plot(ETA,F2,'r','LineWidth',3)
% hold on
end
get(0,'DefaultAxesColorOrder');
plot(ETA,AF2,'LineWidth',2);
legend('b=-3','b=0','b=0.3','b=0.6')
ylabel('Scaled Axial Velocity'); xlabel('Scaled Distance From the Wall')



grid on
 title('Faulkner-Scan Solution')
```

Stability of an imperfect Von Mises' truss

```
clc; clear all; close all

%Input Data
%==========
L=1; AL0=pi/4; a=L*cos(AL0); w0=0;
tan0=tan(AL0);  cos0=cos(AL0);
a1=(pi*w0/(2*L))^2;
A=0.5^2; J=0.5^4/12;
a2=A*L^2/(pi^2*J);

%Initial data
AL(1)=AL0; AP(1)=0;  V(1)=0;%displacement
dsig=0.001;
for i=2:1800
   %Derivatives
   s=sin(AL(i-1)); c=cos(AL(i-1));  p=AP(i-1);
   FP=1+2*a1*a2*s^3/(s-a2*p)^3;  %Derivative by P
   NUM1=3*s^2*c*(s-a2*p)^2;
   NUM2=s^3*2*(s-a2*p)*c;
   DEN=(s-a2*p)^4;
   FAL=a1*(NUM1-NUM2)/DEN+cos(AL0)/c^2-c;  %derivative by ALFA
```

```
    ALD=-FP/sqrt(FP^2+FAL^2);
    PD=FAL/sqrt(FAL^2+FP^2);
    AL(i)=AL(i-1)+ALD*dsig;
    AP(i)=AP(i-1)+PD*dsig;
    V(i)=L*(sin(AL0)-s);
end



plot(V,AP,'linewidth', 4); grid on
xlabel('Vertical Displacement'); ylabel('Force');
title('MISES TRUSS')



hold on
% clear V AP
AP1(1)=0;  V1(1)=0;%displacement
EE(1)=0; %potential energy in the absense of load
for i=2:200
    %Derivatives
    s=sin(AL(i-1)); c=cos(AL(i-1));  p=AP1(i-1);
    FP=1+2*a1*a2*s^3/(s-a2*p)^3;
    NUM1=3*s^2*c*(s-a2*p)^2;
    NUM2=s^3*2*(s-a2*p)*c;
```

```
    DEN=(s-a2*p)^4;
    FAL=a1*(NUM1-NUM2)/DEN+cos(AL0)/c^2-c;
    ALD=FP/sqrt(FP^2+FAL^2);
    PD=-FAL/sqrt(FAL^2+FP^2);
    AL(i)=AL(i-1)+ALD*dsig;
    AP1(i)=AP1(i-1)+PD*dsig;
    V1(i)=L*(sin(AL0)-s);
end
plot(V1,AP1,'linewidth', 4); grid on

hold on

%Total Potential energy
%============================
for i=2:1800
    %Derivatives
    s=sin(AL(i-1)); c=cos(AL(i-1));  p=AP(i-1);
    FP=1+2*a1*a2*s^3/(s-a2*p)^3;  %Derivative by P
    NUM1=3*s^2*c*(s-a2*p)^2;
    NUM2=s^3*2*(s-a2*p)*c;
    DEN=(s-a2*p)^4;
    FAL=a1*(NUM1-NUM2)/DEN+cos(AL0)/c^2-c;  %derivative by ALFA
    ALD=-FP/sqrt(FP^2+FAL^2);
```

```
    PD=FAL/sqrt(FAL^2+FP^2);
    AL(i)=AL(i-1)+ALD*dsig;
    AP(i)=AP(i-1)+PD*dsig;
    V(i)=L*(sin(AL0)-s);
     %Potential energy
     Pbar=0.1;
    EE(i)=(1-cos0*sqrt(tan0^2+(1-V(i)/a*tan0)^2))^2-Pbar*V(i)/L;
end


plot(V,EE,'r','linewidth',4); grid on
xlabel('Vertical Displacement'); ylabel('TOTAL POTENTIAL ENERGY/FORCE');
```

Rotating Disc Boundary Layer Flow

```
% Solving Flow around Rotating disk

clc; close all; clear all
N=100; %number of layers from eta=0 to eta=10 (inf)
Ntau=200; %100; %number of tau imcrements
L=10;  %length
dtau=1/(Ntau-1); %step n load increments
deta=L/N;
% at initial state (tau=0)
ETA=linspace(0,10,N+1);

%Vectors initialization
X1=zeros(N+1,1); X2=zeros(N+1,1); X3=zeros(N+1,1); %X3(1)=tau - corrected inside loop
X4=zeros(N+1,1); X5=zeros(N+1,1); P=zeros;

%Sensitivities (bar variables)
X1B=zeros(N+1,1); X2B=zeros(N+1,1); X3B=zeros(N+1,1);
X4B=zeros(N+1,1); X5B=zeros(N+1,1);

%3 componenets of each sensitivity
X1B0=zeros(N+1,1); X2B0=zeros(N+1,1); X3B0=zeros(N+1,1);
```

```
X4B0=zeros(N+1,1); X5B0=zeros(N+1,1);

X1B1=zeros(N+1,1); X2B1=zeros(N+1,1); X3B1=zeros(N+1,1);
X4B1=zeros(N+1,1); X5B1=zeros(N+1,1);

X1B2=zeros(N+1,1); X2B2=zeros(N+1,1); X3B2=zeros(N+1,1);
X4B2=zeros(N+1,1); X5B2=zeros(N+1,1);


%5 elements components at current node
Z0=zeros(5,1); Z1=zeros(5,1); Z2=zeros(5,1);

%Fixed elements of transfer matrix A
A=zeros(5);
A(1,2)=1; A(3,4)=1; A(5,1)=-2;

for itau=1:Ntau
   %Initial condition for X3  until it reaches 1 - real BC
   X3(1)=(itau-1)*dtau;
   %Initial conditions for components Z0 Z1 Z2
   Z0=[0;0;1;0;0];
   Z1=[0;1;0;0;0];
   Z2=[0;0;0;1;0];
```

```
%Pass to 1D sensitivities componenets
X1B0(1)=Z0(1); X2B0(1)=Z0(2); X3B0(1)=Z0(3);
X4B0(1)=Z0(4); X5B0(1)=Z0(5);

X1B1(1)=Z1(1); X2B1(1)=Z1(2); X3B1(1)=Z1(3);
X4B1(1)=Z1(4); X5B1(1)=Z1(5);

X1B2(1)=Z2(1); X2B2(1)=Z2(2); X3B2(1)=Z2(3);
X4B2(1)=Z2(4); X5B2(1)=Z2(5);

for i=1:N  %integrate eq for sensitivities U and W
   %Variable elements of transfer matrix A
   A(2,:)=[2*X1(i) X5(i) -2*X3(i) 0     X2(i)];
   A(4,:)=[2*X3(i) 0      2*X1(i) X5(i) X4(i)];
   %Integrate sensitivities
   M=eye(5)-deta*A;

   DINV=inv(M);
   Z0=DINV*Z0;
   Z1=DINV*Z1;
   Z2=DINV*Z2;

   %Pass to 1D sensitiviries componenets
```

```
    X1B0(i+1)=Z0(1); X2B0(i+1)=Z0(2); X3B0(i+1)=Z0(3);
    X4B0(i+1)=Z0(4); X5B0(i+1)=Z0(5);

    X1B1(i+1)=Z1(1); X2B1(i+1)=Z1(2); X3B1(i+1)=Z1(3);
    X4B1(i+1)=Z1(4); X5B1(i+1)=Z1(5);

    X1B2(i+1)=Z2(1); X2B2(i+1)=Z2(2); X3B2(i+1)=Z2(3);
    X4B2(i+1)=Z2(4); X5B2(i+1)=Z2(5);
end %i
X1B0; X2B0, X3B0, X4B0; X5B0
X1B1, X2B1, X3B1; X4B1, X5B1;
X1B2; X2B2, X3B2, X4B2; X5B2

%Calculate blend coefficients alfa and beta
RIGHT=-[Z0(1); Z0(3)];
MATR=[Z1(1)  Z2(1); Z1(3) Z2(3)];
SOLUT=MATR\RIGHT;
SOLUT;
alfa=SOLUT(1); beta=SOLUT(2);
%Total sensitivities
X1B=X1B0+alfa*X1B1+beta*X1B2;
X2B=X2B0+alfa*X2B1+beta*X2B2;
X3B=X3B0+alfa*X3B1+beta*X3B2;
```

```
    X4B=X4B0+alfa*X4B1+beta*X4B2;
    X5B=X5B0+alfa*X5B1+beta*X5B2;
    %Update solutions
    X1=X1+dtau*X1B;
    X2=X2+dtau*X2B;
    X3=X3+dtau*X3B;
    X4=X4+dtau*X4B;
    X5=X5+dtau*X5B;
end  %itau
X1
X2
X3
X4
X5;

plot(ETA,X1,'r','LineWidth',3)
grid on; xlabel('ETA'); ylabel('F,G,H'); title('Velocity Distribution');

hold on
plot(ETA,X2,'b','LineWidth',3)
grid on; xlabel('ETA'); ylabel('F,G,H');title('Velocity Distribution');

hold on
```

```
plot(ETA,X3,'k','LineWidth',3)
grid on; xlabel('ETA'); ylabel('F,G,H');title('Velocity Distribution');

hold on
plot(ETA,X4,'y','LineWidth',3)
grid on; xlabel('ETA'); ylabel('F,G,H'); title('Velocity Distribution')

hold on
plot(ETA,-X5,'m','LineWidth',3); title('Velocity Distribution')
grid on; xlabel('ETA'); ylabel('F,G,H');
```

Troesch's Problem

```
function TROESH
clc; close all; clear all
%Simple parameter continuation
N=500; h=1/(N-1); %number of nodes and a step
tau=0; dtau=0.1;
X=linspace(0,1,N);
Y=X;  %Initial approach
for ii=1:1000
   tau=tau+dtau;
   %Coefficients of ODE
   a=-tau^2*cosh(tau*Y)';
   r=(sinh(tau*Y)+tau*cosh(tau*Y).*Y)';
   %Thomas coefficients initialization
   A=ones(N,1); B=ones(N,1); C=ones(N,1);  R=ones(N,1);
   A=A/h^2; C=C/h^2; B=-2*B/h^2+a; R=r;
   A(1)=0; C(1)=0; R(1)=0; B(1)=1;
   A(N)=0; C(N)=0; B(N)=1; R(N)=0;
   %sensitivities
   YB=THOMAS(A,B,C,R);
   %UPDATE
   Y=Y+YB*dtau;
```

```
end
plot(X,Y); grid on
tau



function x=THOMAS(a,b,c,d);
%a, b, c are the column vectors for the compressed tridiagonal matrix, d is the right vector
n = length(a); % n is the number of rows

% Modify the first-row coefficients
al(1) = -c(1) / b(1);    % Division by zero risk.
bt(1) = d(1) / b(1);

for i = 2:n
   temp = b(i) + a(i) * al(i-1);
   al(i) = - c(i) / temp;
   bt(i) = (d(i) - a(i) * bt(i-1))/temp;
end

x(n) = bt(n);

% Now back substitute.
```

```
for i = n-1:-1:1
    x(i) = bt(i) + al(i) * x(i + 1);
end
```

Invariant Imbedding

```
function TROESH

clc; close all; clear all

%Invariant Imbedding based on dicrete scheme

N=100;  %number of nodes


tau=10; lmax=1;


X=linspace(0,1,N);

Y0=X'; Y=Y0; %Initial approach


l=0; dl=0.001;

%A=ones(N,1); B=ones(N,1); C=ones(N,1);  R=ones(N,1);

 A=ones(N,1); A(1)=0; C(1)=0;

 C=ones(N,1); C(N)=0; A(N)=0;

for ii=1:100000  %number of incremental steps


    B=-2*ones(N,1)-tau^2/N^2*l*cosh(tau*Y0);

    B(1)=1; B(N)=1;

    R=tau/N^2*sinh(tau*Y0)';

    R(1)=0; R(N)=0;
```

```
%derivatives at initial point (l=0)
YB1=(THOMAS(A,B,C,R))';
%rough estimate derivative for the 1st interval
size(Y0)
size(YB1)
Y1=Y0+YB1*dl;
'length(Y1)',length(Y1)

'Y1(end)',Y1(end)


l=l+dl;
%Derivative estimation at the end of interval
B=-2*ones(N,1)-tau^2/N^2*l*cosh(tau*Y1);
R=tau/N^2*sinh(tau*Y1)';

%derivatives at end point (l=0)
YB2=THOMAS(A,B,C,R)';
%Averaged
YB=0.5*(YB1+YB2);
%Reached Y at l=l+dl
Y=Y+YB*dl;
%Go loop
```

```
    Y0=Y;
    if(l>lmax), break, end
end
Y(N)=1;
plot(X,Y); grid on
ylim([-0.1 1.1])
xlim([0 1.1])
tau,l
Y
```

```
function x=THOMAS(a,b,c,d);
%a, b, c are the column vectors for the compressed tridiagonal matrix, d is the right vector
n = length(a); % n is the number of rows

% Modify the first-row coefficients
al(1) = -c(1) / b(1);    % Division by zero risk.
bt(1) = d(1) / b(1);

for i = 2:n
    temp = b(i) + a(i) * al(i-1);
    al(i) = - c(i) / temp;
```

```
    bt(i) = (d(i) - a(i) * bt(i-1))/temp;
end

x(n) = bt(n);

% Now back substitute.

for i = n-1:-1:1
    x(i) = bt(i) + al(i) * x(i + 1);
end
```